ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

μΠλ ∀
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

Διπλωματική Εργασία
Constraint Optimization – Bucket Elimination

Κωστόπουλος Δημήτριος
Αριθμός Μητρώου: 200410

Επιβλέπων Καθηγητής
Σταματόπουλος Παναγιώτης

Αθήνα
Σεπτέμβριος 2008

Η παρούσα διπλωματική εργασία
εκπονήθηκε στα πλαίσια των σπουδών
για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης
στη
Λογική και Θεωρία Αλγορίθμων και Υπολογισμού
που απονέμει το
Τμήμα Μαθηματικών
του
Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών

Εγκρίθηκε την χμμμ από Εξεταστική Επιτροπή
αποτελούμενη από τους

Ονοματεπώνυμο	**Βαθμίδα**	**Υπογραφή**
Π. Σταματόπουλος (επιβλέπων)	Επίκουρος Καθηγητής Τμήματος Πληροφορικής και Τηλεπικοινωνιών Ε.Κ.Π.Α.	
Κ. Δημητρακόπουλος	Καθηγητής Τμήματος Μεθοδολογίας, Ιστορίας και Θεωρίας της Επιστήμης Ε.Κ.Π.Α.	
Π. Ροντογιάννης	Αναπληρωτής Καθηγητής Τμήματος Πληροφορικής και Τηλεπικοινωνιών Ε.Κ.Π.Α.	

Στη Βίκυ, που μας άφησε νωρίς.

Περιεχόμενα

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω όλους όσους βοήθησαν στο να γίνει αυτή η διπλωματική εργασία πραγματικότητα. Πρώτα, θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας, τον κύριο Παναγιώτη Σταματόπουλο, που παρόλο τις δύσκολες συνθήκες συνεργασίας μας – ήμουν στα Ψαρρά ένα χρόνο – ήταν πάντα έτοιμος να με βοηθήσει σε ότι πρόβλημα μου παρουσιαζόταν. Με κατάρτισε πάνω στο θέμα της εργασίας και χωρίς τη βοήθειά του δε θα τα είχα καταφέρει. Επίσης θέλω να ευχαριστήσω και τα άλλα δύο μέλη της τριμελούς επιτροπής, τον κύριο Κώστα Δημητρακόπουλο και τον κύριο Παναγιώτη Ροντογιάννη, που δέχθηκαν να αξιολογήσουν την εργασία αυτή. Θα ήταν παράλειψη να μην ευχαριστήσω τους γονείς μου που με στηρίζουν τόσα χρόνια με την αγάπη τους. Τον αδερφό μου Γιώργο, που αν και δε διάβασε την εργασία αυτή (είχε διαβάσει τη πτυχιακή μου εργασία), με στήριξε όλο το τελευταίο διάστημα της εργασίας. Θέλω να πω και ένα ευχαριστώ στους φίλους μου, που με βοήθησαν και αυτοί με το να μη με προτρέπουν να πηγαίνουμε συνέχεια βόλτες, αλλά με το να με σπρώχνουν στη συγγραφή της εργασίας. Τέλος, θέλω να ευχαριστήσω το Θεό που με έχει καλά και μπορώ να εργάζομαι και να μελετώ με το 100% των δυνατοτήτων μου.

1 Εισαγωγή

Όταν ο άνθρωπος ξεκίνησε να βάζει στη ζωή του τα μαθηματικά, τα προβλήματα που είχε να λύσει ήταν απλά. Αργότερα, παρουσιάστηκαν προβλήματα, τα οποία δεν μπορούσε να τα λύσει εύκολα.

Πολλά τέτοια προβλήματα ανήκουν στο NP ή ακόμα και στο #P. Μη γνωρίζοντας, εάν P = NP, δεν μπορούμε να βρούμε σε γρήγορο χρόνο λύση προβλήματος που να ανήκει στο NP. Σε αυτή την εργασία, θα ασχοληθούμε με τέτοια προβλήματα που, χωρίς ακόμα να έχει αποδειχθεί (P $\neq$ NP),πιστεύουμε ότι είναι καταδικασμένα να τρέχουν σε εκθετικό χρόνο, εάν θέλουμε να πάρουμε μια σωστή λύση τους.

Συγκεκριμένα, θα επεξεργαστούμε προβλήματα που έχουν σαν στόχο τους τη μεγιστοποίηση ή ελαχιστοποίηση μιας συνάρτησης. Οπότε το πρόβλημα δεν είναι μόνο εάν θα λύσουμε εύκολα τα προβλήματα αυτά, αλλά αν θα τα λύσουμε σωστά και αν η λύση που θα βρούμε είναι η καλύτερη.

Ευτυχώς, υπάρχουν πολλοί αλγόριθμοι που επιλύουν τα προβλήματα αυτά, σχετικά γρήγορα και με μικρό εύρος λάθους. Μια τέτοια οικογένεια αλγορίθμων θα μελετήσουμε στην εργασία αυτή, τους αλγόριθμους απαλοιφής με κάδους ή όπως αλλιώς ονομάζονται αλγόριθμοι απαλοιφής μεταβλητών. Εγώ προτιμώ τη δεύτερη ονομασία, αλλά η πρώτη έχει επικρατήσει στο χώρο.

Η πορεία μας στην εργασία είναι η εξής: θα ξεκινήσουμε στο δεύτερο κεφάλαιο με την περιγραφή των προβλημάτων ικανοποίησης περιορισμών και τη μοντελοποίησή τους. Στο τρίτο κεφάλαιο, θα περάσουμε στα προβλήματα βελτιστοποίησης με περιορισμούς και θα δούμε τη μοντελοποίησή τους. Στο τέταρτο κεφάλαιο θα δούμε τη διάσημη απαλοιφή με κάδους, αλγόριθμο επίλυσης προβλημάτων βελτιστοποίησης με περιορισμούς. Στο πέμπτο κεφάλαιο, θα δούμε ένα άλλον αλγόριθμο, την απαλοιφή με μικρο-κάδους, που μοιάζει με την απαλοιφή με κάδους. Δε βρίσκει τη βέλτιστη λύση, αλλά εγγυάται ένα διάστημα μέσα στο οποίο θα υπάρχει η λύση αυτή. Στο έκτο κεφάλαιο θα ταξιδέψουμε σε γνωστά προβλήματα όπου η απαλοιφή κάδων είναι καταλυτική για τη λύση τους. Τέλος, στο έβδομο κεφάλαιο, θα δούμε μεθόδους με τις οποίες μπορούμε να κάνουμε τους αλγόριθμους με μικρο-κάδους πιο αποδοτικούς.

Στο τέλος των πιο πολλών κεφαλαίων, υπάρχει μια βιβλιογραφική αναφορά η οποία σχετίζεται με το θέμα των κεφαλαίων. Πριν συνεχίσουμε με το πρώτο κεφάλαιο είναι καλό να αναφέρουμε τα τρία βιβλία που υπάρχουν στο χώρο [Rin03], [RBT06] και [Tsa93], με το πρώτο να είναι το καλύτερο κατά τη γνώμη μου. Ωστόσο, το τρίτο διατίθεται δωρεάν από την ιστοσελίδα http://cswww.essex.ac.uk/Research/CSP/edward/FCS.html.

2 Προβλήματα ικανοποίησης περιορισμών

Στην εργασία αυτή θα ασχοληθούμε με προβλήματα ικανοποίησης περιορισμών (constraint satisfaction problems). Στη βιβλιογραφία, τα προβλήματα αυτά, τα βρίσκουμε με τον όρο CSP που προκύπτει από τα αρχικά γράμματα των αγγλικών λέξεων constraint satisfaction problems. Ας δούμε ένα τέτοιο πρόβλημα.

2.1 Το πρόβλημα της δημοπρασίας

Θα αναφερόμαστε συνέχεια στο πρόβλημα της δημοπρασίας. Είναι προτιμότερο να ξεκινήσουμε με την περιγραφή αυτού του προβλήματος, καθώς θα μας βοηθήσει στην καλύτερη αφομοίωση των ορισμών που θα επακολουθήσουν.

Πριν ξεκινήσουμε, πρέπει να διευκρινίσουμε ότι με τον όρο δημοπρασία θα αναφερόμαστε σε ένα σύνολο προϊόντων. Στο πρόβλημα της δημοπρασίας (auction problem) υπάρχει ένας δημοπράτης (auctioneer) που έχει βγάλει προϊόντα στο σφυρί. Παρατηρούμε ότι δημοπρασίες μπορεί να έχουν κοινά προϊόντα. Ο δημοπράτης δημιουργεί τις δημοπρασίες που τον ενδιαφέρουν και το πρόβλημα έγκειται στο να βρούμε όλα τα σύνολα δημοπρασιών, τέτοια που δημοπρασίες στο ίδιο σύνολο δεν έχουν κοινά προϊόντα.

Για χάριν συντομίας, θα αναφερόμαστε στο πρόβλημα της δημοπρασίας με το ακρωνύμιο AP που σχηματίζεται από τους αγγλικούς όρους auction problem.

2.1.1 Στιγμιότυπο του AP

Παραθέτουμε ένα στιγμιότυπο του AP, το οποίο και ονομάζουμε A. Έχουμε:

- τα προϊόντα $S = \{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8\}$
- τις δημοπρασίες $B = \{b_1,b_2,b_3,b_4,b_5\}$, όπου
  - $b_1 = \{a_1,a_2,a_3,a_4\}$
  - $b_2 = \{a_2,a_3,a_6\}$
  - $b_3 = \{a_1,a_4,a_5\}$
  - $b_4 = \{a_2,a_8\}$
  - $b_5 = \{a_5,a_6\}$

2.2 Δίκτυα με περιορισμούς

Έστω ότι θέλουμε να μοντελοποιήσουμε το AP. Για να το επιτύχουμε αυτό, θα χρησιμοποιήσουμε τα δίκτυα με περιορισμούς.

Ορισμός: Δίκτυο με περιορισμούς

> *Ένα δίκτυο με περιορισμούς (constraint network) είναι η συνολο-τριάδα (X,D,C). Το σύνολο X συμβολίζει τις μεταβλητές του προβλήματος, το σύνολο D είναι το πεδίο από όπου παίρνουν τιμή οι μεταβλητές και το σύνολο C είναι οι περιορισμοί του προβλήματος. Περιορισμός (constraint) μεταξύ μερικών μεταβλητών του προβλήματος είναι μια σχέση μεταξύ των πεδίων τιμών των μεταβλητών. Δηλαδή, ένας περιορισμός ορίζει τις δυνατές αποτιμήσεις από τα πεδία τιμών των μεταβλητών που έχουν κάποια σχέση μεταξύ τους. Γράφουμε,* $C_{x_i,\dots,x_i} \subseteq D_i x \dots x D_j$, *με* $C_{x_i,\dots,x_i} \in C$.

Στόχος είναι να βρούμε όλες τις αποτιμήσεις $\bar{a} = (a_{1,\dots}, a_n)$, έτσι ώστε το πρόβλημα να έχει λύση, με $a_i \varepsilon D_i$.

Ορισμός: Συνεπές – Ασυνεπές πρόβλημα

Πρόβλημα που δεν έχει λύση καλείται ασυνεπές (inconsistent), αλλιώς συνεπές (consistent).

2.2.1 Δίκτυο με περιορισμούς για το AP

Ας μοντελοποιήσουμε το πρόβλημά μας για να καταλάβουμε καλύτερα την έννοια του δικτύου με περιορισμούς.

- Το σύνολο X είναι οι δημοπρασίες που επιλέγει ο δημοπράτης.
- D είναι το δισύνολο {0,1}, όπου με 0 συμβολίζουμε εάν ο δημοπράτης δεν πρέπει να επιλέξει τη δημοπρασία αυτή και με 1 διαφορετικά.
- Το σύνολο C είναι οι περιορισμοί του προβλήματος, ότι οι δημοπρασίες δεν πρέπει να περιέχουν κοινά προϊόντα.

2.2.2 Δίκτυο με περιορισμούς για στιγμιότυπο του AP

Θα χρησιμοποιήσουμε το στιγμιότυπο A που έχουμε περιγράψει. Για να έχουμε δίκτυο με περιορισμούς, μένει να περιγράψουμε τους περιορισμούς του προβλήματος. Πρέπει δύο δημοπρασίες που έχουν ένα κοινό προϊόν να μην είναι και οι δύο επιλεγμένες, να μην έχουν και οι δύο αποτιμηθεί με την τιμή 1.

Για παράδειγμα, οι δημοπρασίες b_1 και b_2 έχουν κοινά τα προϊόντα a_1 και a_2. Δεν πρέπει να πάρουν και οι δύο μαζί την τιμή 1 και έχουμε $R_{12} = \{(1,0),(0,1),(0,0)\}$.

Όλοι οι περιορισμοί:

- $C = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12} = R_{13} = R_{14} = R_{24} = R_{25} = R_{35} = \{(1,0),(0,1),(0,0)\}$.
- Δεν πρέπει να ξεχάσουμε να αναφέρουμε τις μεταβλητές:
  $X = \{x_1,x_2,x_3,x_4,x_5\}$, όπου το x_i συμβολίζει τη δημοπρασία b_i
- και τα πεδία τιμών:
  $D = \{D_1,D_2,D_3,D_4,D_5\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = \{0,1\}$, όπου D_i είναι το πεδίο τιμών της μεταβλητής x_i.

2.3 Βασικές θεωρητικές γνώσεις

Στα επόμενα κεφάλαια θα ασχοληθούμε με δίκτυα με κόστη. Τα δίκτυα αυτά δε διαφέρουν κατά πολύ από τα δίκτυα με περιορισμούς. Ωστόσο, η μελέτη των δικτύων με κόστη θα μας απομακρύνει από τη μελέτη των δικτύων με περιορισμούς. Αυτό όμως δεν είναι καλό γιατί για δίκτυα με περιορισμούς έχουν γίνει πολλές μελέτες και έχουν προκύψει πολλά αποτελέσματα, τα οποία μπορούν να χρησιμοποιηθούν και στα δίκτυα με κόστη. Αν και δεν είναι απαραίτητο για να μελετήσει κάποιος αυτή την εργασία, είναι σημαντικό, εάν όχι απαραίτητο, ο αναγνώστης αυτής της διπλωματικής εργασίας να μελετήσει και το υπόλοιπο κομμάτι αυτού του κεφαλαίου. Υπάρχουν ενδιαφέροντα αποτελέσματα πάνω σε δίκτυα με περιορισμούς. Εάν κάποιος θέλει να έχει μια σφαιρική άποψη για την περιοχή πρέπει να διαβάσει τα επόμενα εδάφια. Αν όχι, θα είναι σαν να μαθαίνει ένα παιδί στο δημοτικό πολλαπλασιασμό χωρίς να ξέρει πρόσθεση.

2.3.1 Συμπερασμός

Έστω ότι έχουμε ένα δίκτυο R με τους εξής περιορισμούς: $R_{12} = x_1 > x_2$ και $R_{23} = x_2 > x_3$, με $x_1 \in \{1,2,3\}$, $x_2 \in \{1,2,3\}$ και $x_3 \in \{1,2,3\}$. Από τις δύο ανισώσεις – περιορισμούς μπορούμε να συμπεράνουμε ότι $x_1 > x_3$ και να δούμε τη νέα ανισότητα σαν μια σχέση – περιορισμό, $R_{13} = x_1 > x_3$. Αυτή η διαδικασία, δημιουργίας νέων περιορισμών από ήδη υπάρχοντες, ονομάζεται συμπερασμός (inference). Υπάρχουν πολλοί τρόποι να επιτύχουμε συμπερασμό και θα δούμε κάποιους από αυτούς.

2.3.2 Ισοδύναμα δίκτυα

Έστω R' το νέο δίκτυο που προκύπτει προσθέτοντας την R_{13} στο R, στο προηγούμενο παράδειγμα που δώσαμε για τον συμπερασμό. Τα δύο δίκτυα είναι ισοδύναμα:

Ορισμός: ισοδύναμα δίκτυα

> *Δύο δίκτυα καλούνται ισοδύναμα (equivalent), εάν είναι ορισμένα πάνω στο ίδιο σύνολο μεταβλητών και έχουν τις ίδιες ακριβώς λύσεις!*

Τα R και R' είναι ισοδύναμα γιατί ορίζονται στις ίδιες μεταβλητές x_1, x_2 και x_3 και έχουν την ίδια λύση $(x_1,x_2,x_3) = (3,2,1)$.

2.3.3 Ελαχιστικά δίκτυα

Γενικά, όλες οι μελέτες και τα αποτελέσματα που έχουν προκύψει στο χώρο της επεξεργασίας των περιορισμών είναι πάνω σε δυαδικά δίκτυα με περιορισμούς (δίκτυα με μόνο δυαδικούς περιορισμούς[1]). Αυτό συνεπάγεται ότι θα θέλαμε να αντιμετωπίζουμε μόνο δυαδικά δίκτυα και όταν αυτό δεν είναι εφικτό να τα μετατρέπουμε σε ένα δυαδικό ισοδύναμο. Δυστυχώς, δεν υπάρχει για όλα τα δίκτυα ένα ισοδύναμο δυαδικό, αλλά μπορούμε να κατασκευάσουμε ένα άλλο δυαδικό που να είναι σχεδόν ισοδύναμο. Αυτή την περιγραφή «σχεδόν ισοδύναμο» θα εξηγήσουμε ευθύς αμέσως.

Ορισμός: «Πιο αυστηρό δίκτυο» σχέση

> *Λέμε ότι ένα δυαδικό δίκτυο R είναι πιο αυστηρό (tighter) από ένα άλλο δίκτυο R', εάν και μόνον εάν τα R και R' είναι ορισμένα στο ίδιο σύνολο μεταβλητών και εάν για κάθε περιορισμό μεταξύ δύο μεταβλητών i και j έχουμε* $R_{ij} \subset R'_{ij}$.

Ακριβώς αυτή η σχέση «πιο αυστηρό δίκτυο», θα μας βοηθήσει για την περιγραφή «σχεδόν ισοδύναμο». Έτσι, αν και όλα τα δίκτυα δεν μπορούμε να κατασκευάσουμε ένα ισοδύναμο δυαδικό, μπορούμε να κατασκευάσουμε ένα δυαδικό που να είναι το πιο αυστηρό δυαδικό δίκτυο που υπάρχει. Αυτή την κατασκευή θα μελετήσουμε ευθύς αμέσως.

Ορισμός: Δίκτυο προβολής μιας σχέσης

> *Έστω ότι έχουμε μία σχέση ρ. Δίκτυο προβολής (projection network) P(ρ) για τη σχέση αυτή, καλούμε το δίκτυο που έχει σαν μεταβλητές, τις μεταβλητές της ρ και σαν περιορισμούς τις σχέσεις που προκύπτουν ως εξής: για κάθε δυνατή δυάδα μεταβλητών παίρνουμε την προβολή της δυάδας αυτής στη ρ και σχηματίζεται με αυτόν τον τρόπο μια νέα σχέση περιορισμός.*

Έστω ότι έχουμε ένα δίκτυο R. Έστω sol(R) το σύνολο των λύσεων αυτού του δικτύου. Το σύνολο αυτών των λύσεων μπορούμε να το εκφράσουμε με μία σχέση ρ. Εάν πάρουμε το δίκτυο προβολής P(ρ), τότε το P(ρ) είναι το πιο αυστηρό δυαδικό δίκτυο που μπορούμε να έχουμε ώστε:

1. Η σχέση ρ ανήκει στο σύνολο των λύσεων του δικτύου προβολής της, $\rho \in \mathrm{Sol}(P(\rho))$. Αυτή είναι ιδιότητα που ισχύει για όλα τα δίκτυα προβολής.
2. Δεν υπάρχει άλλο δυαδικό δίκτυο R' τέτοιο που η ρ να ανήκει στο σύνολο των λύσεων του R' και το σύνολο των λύσεων του R' να είναι γνήσιο υποσύνολο των λύσεων του P(ρ), $\neg \exists R' : \rho \subseteq \mathrm{sol}(R') \subset \mathrm{sol}(P(r))$.

1 Για την ακρίβεια, στα δυαδικά δίκτυα επιτρέπουμε και περιορισμούς ορισμένους πάνω σε μία μόνο μεταβλητή, αλλά τέτοιοι περιορισμοί μόνο ενδιαφέρον δεν έχουν, καθώς ουσιαστικά ορίζουν εκ νέου τη μεταβλητή με ένα νέο μικρότερο σε πληθικότητα πεδίο ορισμού.

Μπορεί να υπάρχουν πολλά δυαδικά ισοδύναμα δίκτυα με ένα δίκτυο R, αλλά λόγω της ιδιότητας 2, το δίκτυο προβολής της λύσης του είναι το πιο αυστηρό. Αυτή ακριβώς την ιδιότητα την μετατρέπουμε σε έναν ορισμό και ένα θεώρημα.

Ορισμός: Ελαχιστικό δίκτυο

> *Έστω ένα δίκτυο R. Ελαχιστικό (minimal) , καλούμε το δίκτυο M(R), που είναι ίσο με την τομή όλων των ισοδύναμων του δικτύων. Συχνά αντί για M(R) συμβολίζουμε το ελαχιστικό δίκτυο με M(ρ), όπου ρ = Sol(R).*

Θεώρημα: Ελαχιστικά δίκτυα για δυαδικά δίκτυα

> *Το ελαχιστικό δίκτυο M(R) = M(ρ) για ένα δυαδικό δίκτυο R είναι το δίκτυο προβολής P(ρ) της λύσης ρ του R, ρ = sol(R). M(ρ) = M(R) = P(ρ). Επίσης τα M(ρ), P(ρ) και R είναι ισοδύναμα.*

2.3.3.1 Παράδειγμα ελαχιστικού δικτύου

Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R με τρεις μεταβλητές x_1, x_2 και x_3 με το ίδιο πεδίο ορισμού D, D = {1,2,3}. Στο δίκτυο έχουμε και το μοναδικό περιορισμό $R_{123} = x_1 > x_2 > x_3$. Η λύση του προβλήματος είναι η $\rho_{123} = (3,2,1)$.

Δημιουργούμε το δίκτυο προβολής P(ρ). Το σύνολο των μεταβλητών παραμένει το ίδιο ενώ οι περιορισμοί είναι τώρα οι $\rho_{12} = (3,2)$, $\rho_{13} = (3,1)$ και $\rho_{23} = (2,1)$. Αυτό το δίκτυο δεν είναι το ελαχιστικό, καθώς το δίκτυο R' με περιορισμούς $\rho'_{12} = (3,2)$ και $\rho'_{23} = (2,1)$ είναι πιο αυστηρό και ισοδύναμο με τα R και P(ρ). Το R' είναι και το ελαχιστικό δίκτυο M(R) του R.

2.3.4 Συνέπεια ακμών

Τα ελαχιστικά δίκτυα έχουν μια καταπληκτική ιδιότητα: κάθε ανάθεση τιμών σε μία μεταβλητή μπορεί να επεκταθεί με μία ανάθεση τιμών σε μία δεύτερη μεταβλητή, διατηρώντας ταυτόχρονα τη συνέπεια. Αυτή την πολύ ωραία ιδιότητα μπορούν να την αποκτήσουν και τα μη-ελαχιστικά δίκτυα, εάν εφαρμόσουμε συνέπεια ακμών (arc consistency).

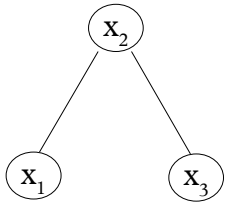
Ορισμός: Συνέπεια ακμών

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Έστω μία μεταβλητή x_i, μια μεταβλητή x_j και ένα περιορισμός R_{ij} που ορίζεται πάνω στις x_i και x_j. Λέμε ότι η μεταβλητή x_i είναι συνεπής ως προς τις ακμές σε σχέση με τη μεταβλητή x_j (arc-consistent relative to), εάν και μόνον εάν για κάθε τιμή a_i του πεδίου ορισμού της x_i υπάρχει τιμή a_j του πεδίου ορισμού της x_j, έτσι ώστε* $(a_i, a_j) \in R_{ij}$.
>
> *Η ακμή $<x_i,x_j>$ είναι συνεπής ως προς τις ακμές (arc-consistent), εάν και μόνον εάν η x_i είναι συνεπής ως προς τις ακμές σε σχέση με τη x_j και εάν η x_j είναι συνεπής ως προς τις ακμές σε σχέση με τη x_i.*
>
> *Το δίκτυο R είναι συνεπές ως προς τις ακμές (arc-consistent), εάν και μόνον αν κάθε ακμή του R, είναι συνεπής ως προς τις ακμές.*

2.3.4.1 Παράδειγμα συνέπειας ακμών

Χάριν συντομίας θα καλούμε τη συνέπεια ακμών σαν AC, από το ακρωνύμιο που

σχηματίζεται από τον αγγλικό όρο arc-consistency. Έστω το δίκτυο R με τους εξής περιορισμούς: $R_{12} = x_1 > x_2$ και $R_{23} = x_2 > x_3$, με $x_1 \in \{1,2,3\}$, $x_2 \in \{1,2,3\}$ και $x_3 \in \{1,2,3\}$. Μετατρέπουμε το δίκτυο σε ένα συνεπές ως προς τις ακμές δίκτυο. Ας σχηματίσουμε το γράφο με περιορισμούς του.



Σχήμα 1: Ο γράφος με περιορισμούς του προβλήματος

Πρέπει να κάνουμε συνεπείς ως προς τις ακμές τις ακμές $<x_1,x_2>$ και $<x_2,x_3>$. Για να κάνουμε συνεπή ως προς τις ακμές την ακμή $<x_1,x_2>$ πρέπει να κάνουμε συνεπή ως προς τις ακμές τη μεταβλητή x_1 σε σχέση με τη μεταβλητή x_2. Εξετάζουμε για κάθε τιμή a_1 της μεταβλητής x_1 εάν υπάρχει τιμή a_2 στο πεδίο ορισμού της μεταβλητής x_2 έτσι ώστε το (a_1,a_2) να ανήκει στη R_{12}. Για τις τιμές 2 και 3 της x_1 υπάρχουν οι τιμές 1 και 2 της x_2 αντίστοιχα. Αλλά για την τιμή 1 της x_1 δεν μπορούμε να βρούμε τιμή για τη x_2 που να δημιουργεί μία συνεπή πλειάδα της R_{12}. Αφαιρούμε την τιμή 1 από τη μεταβλητή x_1.

Ομοίως, αφαιρούμε την τιμή 3 από τη x_2. Για να εφαρμόσουμε AC στη $<x_2,x_3>$ αφαιρούμε την τιμή 1 από τη x_2 και τις τιμές 2 και 3 από τη x_3. Δεν τελειώσαμε, η AC για την $<x_2,x_3>$ αφαίρεσε την τιμή 1 από τη x_2, οπότε ελέγχουμε πάλι για AC για τη $<x_1,x_2>$ και αφαιρούμε την τιμή 2 από τη μεταβλητή x_1. Τελικά έχουμε $x_1 = \{3\}$, $x_2 = \{2\}$, $x_3 = \{1\}$.

2.3.4.2 Αλγόριθμοι και πολυπλοκότητα για συνέπεια ακμών.

Η εφαρμογή της AC, στο προηγούμενο παράδειγμα, ήταν ουσιαστικά η περιγραφή του πιο απλού αλγόριθμου για την εφαρμογή AC σε ένα δίκτυο με περιορισμούς.

Ο αλγόριθμος REVISE(x_i,x_j) αφαιρεί τιμές από το πεδίο τιμών της μεταβλητής x_i, έτσι ώστε η x_i να είναι συνεπής ως προς τις ακμές σε σχέση με τη μεταβλητή x_j. Η πολυπλοκότητά του είναι $O(k^2)$, όπου k είναι η πληθικότητα του μεγαλύτερου πεδίου τιμών των μεταβλητών.

Ο αλγόριθμος AC-1(R) εφαρμόζει AC σε ένα δίκτυο με περιορισμούς R με τον τρόπο που περιγράψαμε στο παράδειγμα. Η πολυπλοκότητά του είναι $O(enk^3)$, όπου n είναι το πλήθος των μεταβλητών και e το πλήθος των δυαδικών περιορισμών.

Άλλοι γνωστοί αλγόριθμοι για AC είναι οι AC-3(R) και AC-4(R) με πολυπλοκότητα $O(ek^3)$ και $O(ek^2)$ αντίστοιχα. Υπάρχει και ο AC-5(R), που για συγκεκριμένα είδη περιορισμών, έχει πολυπλοκότητα O(ek).

2.3.5 Συνέπεια μονοπατιού

Η AC μπορεί να αποφασίσει ασυνέπεια. Ας δούμε ένα απλό παράδειγμα. Έστω το δίκτυο R με τους εξής περιορισμούς: $R_{12} = x_1 > x_2$ και $R_{23} = x_2 > x_3$, με $x_1 \in \{1,2\}$, $x_2 \in \{1,2\}$ και $x_3 \in \{1,2\}$. Η AC θα αφαιρέσει την τιμή 1 από το πεδίο ορισμού της x_1 και την τιμή 2 από την x_2 για τον περιορισμό R_{12}. Για τον περιορισμό R_{23} θα αφαιρέσει την τιμή 1 από το πεδίο ορισμού της x_2 και η μεταβλητή x_2 θα μείνει χωρίς τιμές! Οπότε στη x_2 δεν μπορεί να ανατεθεί μία τιμή και έχουμε ασυνέπεια. Η AC μας γλύτωσε από τον κόπο να λύσουμε με μία μέθοδο το δίκτυο, επειδή βρήκε ότι το δίκτυο είναι ασυνεπές.

Μπορεί στο παραπάνω παράδειγμα να βρήκε η AC ότι το δίκτυο είναι ασυνεπές, αλλά αυτό δε συνεπάγεται ότι η AC μπορεί να ανακαλύπτει ασυνέπεια σε οποιοδήποτε δίκτυο. Ορίστε ένα

παράδειγμα όπου το δίκτυο είναι ασυνεπές και η AC δεν μπορεί να το αποφασίσει.

Έστω το δίκτυο R με τους εξής περιορισμούς: $R_{12}=x_1\neq x_2$, $R_{13}=x_1\neq x_3$ και $R_{23}=x_2\neq x_3$, με $x_1\in\{1,2\}$, $x_2\in\{1,2\}$ και $x_3\in\{1,2\}$. Εφαρμόζοντας AC δε θα αλλάξει το πεδίο ορισμού καμίας μεταβλητής. Το δίκτυο είναι ασυνεπές και η AC δεν το βρίσκει. Μπορούμε να ανακαλύψουμε την ασυνέπεια, «επεκτείνοντας» την AC σε τρεις μεταβλητές!

Ορισμός: Συνέπεια μονοπατιού

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Έστω ότι έχουμε τρεις μεταβλητές x_i, x_j και x_k. Το δισύνολο $\{x_i,x_j\}$ είναι συνεπές ως προς το μονοπάτι σε σχέση με τη μεταβλητή x_k ($\{x_i,x_j\}$ is path-consistent relative to x_k), εάν και μόνον εάν για κάθε συνεπή πλειάδα (a_i,a_j) του περιορισμού R_{ij} υπάρχει τιμή a_k της μεταβλητής x_k έτσι ώστε οι πλειάδες (a_i,a_k) και (a_i,a_k), των περιορισμών R_{ik} και R_{jk} αντίστοιχα, είναι συνεπείς. Ισοδύναμα λέμε ότι ο δυαδικός περιορισμός R_{ij} είναι συνεπής ως προς το μονοπάτι σε σχέση με τη μεταβλητή x_k (R_{ij} is path-consistent relative to x_k).*
>
> *Το δίκτυο R είναι συνεπές ως προς το μονοπάτι (path-consistent), εάν και μόνον εάν κάθε δυαδικός περιορισμός του R_{ij}, συμπεριλαμβανομένων των καθολικών περιορισμών, είναι συνεπής ως προς το μονοπάτι σε σχέση με κάθε μεταβλητή που ανήκει στο $S - \{x_i,x_j\}$.*

2.3.5.1 Παράδειγμα συνέπειας μονοπατιού

Αξίζει να αναφέρουμε ότι στη βιβλιογραφία, η συνέπεια μονοπατιού είναι γνωστή σαν PC από το ακρωνύμιο που σχηματίζεται από τις λέξεις path-consistency. Επιστρέφουμε στο παράδειγμα όπου η AC δεν μπορούσε να αποφασίσει ασυνέπεια. Έστω το δίκτυο R με τους εξής περιορισμούς: $R_{12}=x_1\neq x_2$, $R_{13}=x_1\neq x_3$ και $R_{23}=x_2\neq x_3$, με $x_1\in\{1,2\}$, $x_2\in\{1,2\}$ και $x_3\in\{1,2\}$. Παίρνουμε τον περιορισμό R_{12} και την μεταβλητή x_3. Η PC θα ανακαλύψει αμέσως ασυνέπεια, καθώς δεν υπάρχουν τιμές a_1, a_2 και a_3 για τις μεταβλητές x_1, x_2 και x_3 έτσι ώστε $(a_1,a_3)\in R_{13}$ και $(a_2,a_3)\in R_{23}$. Η PC βρήκε ότι το R είναι ασυνεπές.

2.3.5.2 Αλγόριθμοι και πολυπλοκότητα για συνέπεια μονοπατιού

Ο αλγόριθμος REVISE-3(x_i,x_j,x_k) βεβαιώνει ότι το δισύνολο $\{x_i,x_j\}$ είναι συνεπές ως προς το μονοπάτι σε σχέση με τη μεταβλητή x_k. Η πολυπλοκότητά του είναι $O(k^3)$, όπου k είναι η πληθικότητα του μεγαλύτερου πεδίου τιμών των μεταβλητών.

Ο αλγόριθμος PC-1(R) εφαρμόζει PC στο δίκτυο R και έχει πολυπλοκότητα $O(n^5k^5)$, με n το πλήθος των μεταβλητών. Όπως και στην AC και στην PC έχει βελτιωθεί η πολυπλοκότητα αυτή και ο αλγόριθμος PC-2(R) έχει πολυπλοκότητα $O(n^3k^5)$. Παραπάνω βελτίωση επιφέρει ο αλγόριθμος PC-4(R) με πολυπλοκότητα $O(n^3k^3)$. Μάλιστα, ο PC-4(R) είναι ο βέλτιστος αλγόριθμος που μπορεί να υπάρξει για PC.

2.3.6 i-συνέπεια

Στο προηγούμενο παράδειγμα, είχαμε δώσει ένα δίκτυο όπου η AC δεν μπορούσε να ανακαλύψει την ασυνέπεια του δικτύου, ενώ η PC μπορούσε. Μπορούμε να βρούμε άλλο ασυνεπές δίκτυο, όπου η PC δεν μπορεί να βρει την ασυνέπεια. Επεκτείνοντας τη PC σε περισσότερες μεταβλητές, μπορεί να είμαστε πιο τυχεροί και βρούμε την ασυνέπεια. Αυτό ακριβώς θα κάνουμε τώρα, θα επεκτείνουμε τη PC από τρεις μεταβλητές σε περισσότερες, σε i μεταβλητές.

Ορισμός: i-συνέπεια

Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Ένας περιορισμός R_S, ορισμένος σε ένα σύνολο μεταβλητών S, πληθικότητας i-1, είναι i-συνεπής σε σχέση με μια μεταβλητή x_j (i-consistent relative to x_j), η οποία δεν είναι μέλος του S, εάν και μόνον εάν για κάθε πλειάδα t του R_S, υπάρχει τιμή a_j της μεταβλητής x_j, τέτοια που η πλειάδα (t,a_j) είναι συνεπής.

Το δίκτυο R είναι i-συνεπές (i-consistent), εάν και μόνον εάν κάθε περιορισμός R_S του δικτύου, που είναι ορισμένος σε i-1 μεταβλητές, είναι i-συνεπής σε σχέση με κάθε μία από τις μεταβλητές του δικτύου που δεν ανήκουν στην εμβέλεια του R_S.

Λέμε ότι το δίκτυο R είναι ισχυρά i-συνεπές (strongly i-consistent), εάν και μόνον εάν είναι και j-συνεπές για κάθε $j \leqslant i$.

Το δίκτυο R είναι καθολικά συνεπές (globally consistent) εάν και μόνον εάν είναι n-συνεπές, όπου n είναι το πλήθος των μεταβλητών.

Παρατηρούμε ότι για δυαδικά δίκτυα η 2-συνέπεια είναι ισοδύναμη με την συνέπεια ακμών και η 3-συνέπεια ισοδύναμη με την συνέπεια μονοπατιού. Επίσης, όταν φτάσουμε σε καθολική συνέπεια, γνωρίζουμε με βεβαιότητα πια ότι το δίκτυο δεν είναι ασυνεπές.

2.3.6.1 Αλγόριθμοι και πολυπλοκότητά για i-συνέπεια

Όπως στις AC και PC, έτσι και στην i-συνέπεια (i-consistency) ο αλγόριθμος REVISE-$i(x_1,...,x_{i-1},x_i)$ εξετάζει εάν το σύνολο $\{x_1,...,x_{i-1}\}$ είναι i-συνεπές σε σχέση με τη μεταβλητή x_i. Η πολυπλοκότητα του είναι $O((2k)^i)$, με k τη μεγαλύτερη πληθικότητα του πεδίου ορισμού των μεταβλητών.

Ένας απλός αλγόριθμος που στηρίζεται στη REVISE-i, έχει πολυπλοκότητα $O(2^i(nk)^{2i})$ σε χρόνο και $O(n^ik^i)$ σε χώρο, με n το πλήθος των μεταβλητών. Πρέπει να προσθέσουμε ότι υπάρχει αλγόριθμος που να επιτυγχάνει $O((nk)^i)$ πολυπλοκότητα, που είναι και το βέλτιστο που μπορούμε να έχουμε.

2.3.7 Κατευθυνόμενη i-συνέπεια

Εάν θέλουμε να ελέγξουμε την ασυνέπεια ενός δικτύου, δεν έχουμε παρά να ελέγξουμε, εάν είναι καθολικά συνεπές. Το να υπολογίσουμε εάν ένα δίκτυο είναι καθολικά συνεπές, μας κοστίζει $O((nk)^n)$, το οποίο είναι πάρα πολύ ακριβό. Για να αποφύγουμε ένα τόσο μεγάλο κόστος, εφαρμόζουμε i-συνέπεια στο δίκτυό μας, ελπίζοντας ότι όντας τυχεροί, θα φτάσουμε σε ασυνέπεια εφόσον το δίκτυο είναι ασυνεπές. Υπάρχει καλύτερη λύση από αυτή που προτείναμε μόλις τώρα. Πριν την παραθέσουμε, ας δούμε μια άλλη ιδιότητα των καθολικά συνεπών δικτύων.

Εάν ένα δίκτυο έχει i-συνέπεια, έχει την εξής ιδιότητα: για κάθε συνεπή αποτίμηση σε i-1 μεταβλητές μπορεί να βρεθεί αποτίμηση για μια ακόμα i-οστή μεταβλητή, έτσι ώστε η συνολική αποτίμηση στις i μεταβλητές να είναι συνεπής. Για ισχυρά i-συνεπές δίκτυο, αυτό συνεπάγεται ότι μπορούμε, ξεκινώντας από το μηδέν, να αναθέτουμε σε μία μία μεταβλητή μια κατάλληλη τιμή έτσι ώστε να φτάσουμε σε συνεπή ανάθεση i μεταβλητών. Για καθολικά συνεπές δίκτυο αυτό επιφέρει λύση του δικτύου σε γραμμικό χρόνο!

Αυτή η ιδιότητα της καθολικής συνέπειας είναι πιο ισχυρή από αυτό που αρχικά καταλαβαίνουμε με τη γραμμική επίλυση του δικτύου. Το δίκτυο δεν επιλύεται απλά σε γραμμικό χρόνο, αλλά επιλύεται σε γραμμικό χρόνο με όποια σειρά και να πάρουμε τις μεταβλητές μας!

Αυτή μπορεί να είναι μία πολύ ωραία ιδιότητα, αλλά δε μπορούμε να πούμε ότι είναι επιθυμητή. Ίσως, εάν είχαμε γραμμική λύση του δικτύου κατά μία μόνο απαρίθμηση των

μεταβλητών μας να χρειαζόμασταν λιγότερο χώρο και χρόνο για να το επιτύχουμε. Ένας τρόπος να βρούμε τη λύση είναι αρκετός, αφού η λύση είναι αυτή που μας ενδιαφέρει. Το να βρούμε τη λύση του δικτύου σε γραμμικό χρόνο κατά μία συγκεκριμένη απαρίθμηση των μεταβλητών εξοικονομώντας χώρο και χρόνο είναι εφικτό!

Ορισμός: Κατευθυνόμενη i-συνέπεια

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Το δίκτυο R είναι κατευθυνόμενα i-συνεπές σε σχέση με μία απαρίθμηση μεταβλητών d (directional i-consistent relative to order d), εάν και μόνον εάν κάθε σύνολο S i-1 μεταβλητών είναι i-συνεπές σε σχέση με κάθε μεταβλητή που είναι πιο δεξιά στη d από τα μέλη – μεταβλητές του S.*
>
> *Ένα δίκτυο είναι ισχυρά κατευθυνόμενα i-συνεπές σε σχέση με μία απαρίθμηση d (strongly directional i-consistent relative to order d), εάν και μόνον εάν είναι κατευθυνόμενα j-συνεπές σε σχέση με τη d, για κάθε $j \leqslant i$.*

Ανάλογα με την AC, έτσι και με την κατευθυνόμενη i-συνέπεια (directional i-consistency) υπάρχει κατευθυνόμενη συνέπεια ορισμένη ειδικά για δύο ή τρεις μεταβλητές:

Ορισμός: Κατευθυνόμενη συνέπεια ακμών

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Το δίκτυο R είναι κατευθυνόμενα συνεπές ως προς τις ακμές σε σχέση μια απαρίθμηση μεταβλητών d (directional arc-consistent relative to order d), εάν και μόνον εάν κάθε μεταβλητή x_i είναι συνεπής ως προς τις ακμές σε σχέση με κάθε μεταβλητή x_j που είναι πιο δεξιά από τη x_i στη d. Δηλαδή, εάν $d = \{x_1,...,x_n\}$, τότε $i < j$.*

Ορισμός: Κατευθυνόμενη συνέπεια μονοπατιού

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Το δίκτυο R είναι κατευθυνόμενα συνεπές ως προς το μονοπάτι σε σχέση με μία απαρίθμηση μεταβλητών d (directional path-consistent relative to order d), εάν και μόνον εάν κάθε δισύνολο μεταβλητών είναι συνεπές ως προς το μονοπάτι σε σχέση με κάθε μεταβλητή που είναι πιο δεξιά στη d από τα μέλη – μεταβλητές του δισυνόλου.*

Για δυαδικά δίκτυα, ισχύει ότι η κατευθυνόμενη συνέπεια ακμών (directional arc consistency) είναι ισοδύναμη με την κατευθυνόμενη 2-συνέπεια και η κατευθυνόμενη συνέπεια μονοπατιού (directional path consistency) είναι ισοδύναμη με την κατευθυνόμενη 3-συνέπεια, λαμβάνοντας υπόψιν σε όλες τις περιπτώσεις την ίδια απαρίθμηση μεταβλητών.

2.3.7.1 Αλγόριθμοι και πολυπλοκότητα για κατευθυνόμενη συνέπεια

Ο αλγόριθμος DAC(R) εφαρμόζει κατευθυνόμενη συνέπεια ακμών, σε σχέση με μία απαρίθμηση μεταβλητών d, με πολυπλοκότητα $O(ek^2)$, με k τη μεγαλύτερη πληθικότητα του πεδίου ορισμού των μεταβλητών και e το πλήθος των δυαδικών περιορισμών.

Ο αλγόριθμος DPC(R) εφαρμόζει κατευθυνόμενη συνέπεια μονοπατιού σε σχέση με μία απαρίθμηση μεταβλητών d με πολυπλοκότητα $O(n^3k^3)$.

Ο αλγόριθμος $DIC_i(R)$ εφαρμόζει κατευθυνόμενη i-συνέπεια σε σχέση με μία απαρίθμηση μεταβλητών d με πολυπλοκότητα $O(n(w^*(d)^i(2k)^i)$, όπου n το πλήθος των μεταβλητών και $w^*(d)$, το πλάτος του επαγόμενου γράφου, του γράφου με περιορισμούς του δικτύου κατά την απαρίθμηση μεταβλητών d. Ο ορισμός του επαγόμενου γράφου, καθώς και το πλάτος του, αναλύονται στην παράγραφο 4.5.1.5.

Ο αλγόριθμος ADC εφαρμόζει κατευθυνόμενη n-συνέπεια σε σχέση με μία απαρίθμηση μεταβλητών d με πολυπλοκότητα $O(n(2k)^{w^*(d)+1})$.

Καταφέραμε να μειώσουμε την πολυπλοκότητα της καθολικής συνέπειας; Η πολυπλοκότητα για καθολική συνέπεια είναι $O((nk)^n)$, ενώ η πολυπλοκότητα για κατευθυνόμενη n-συνέπεια είναι $O(n(2k)^{w^*(d)+1})$. Δεδομένου ότι $w^*(d) \leqslant n$, το έχουμε καταφέρει!

2.3.8 Σχεσιακή συνέπεια

Όλα τα είδη των συμπερασμών – συνεπειών που έχουμε δει έως τώρα, εξαρτώνται από το πλήθος των μεταβλητών του δικτύου. Υπάρχουν και άλλα είδη συμπερασμών – συνεπειών, που εξαρτώνται από το πλήθος των περιορισμών στο δίκτυο. Ανάλογα με τη φύση του δικτύου, εφαρμόζουμε και το κατάλληλο είδος συμπερασμού – συνέπειας.

Ας δούμε το αντίστοιχο ορισμό της συνέπειας ακμών, τη σχεσιακή συνέπεια ακμών (relational arc consistency):

Ορισμός: Σχεσιακή συνέπεια ακμών

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Έστω ένας περιορισμός R_S ορισμένος σε ένα σύνολο μεταβλητών S. Έστω μία τυχαία μεταβλητή* $x_i \in S$ *και ένας περιορισμός* $R_{S-\{x_i\}}$ *ορισμένος στο S-{x_i}. Ο R_S είναι σχεσιακά συνεπής ως προς τις ακμές σε σχέση με τη x_i (relational arc-consistent relative to x_i), εάν και μόνον εάν ο* $R_{S-\{x_i\}}$ *μπορεί να επεκταθεί στον R_S για κάποια τιμή* $a_{x_i} \in D_{x_i}$ *της μεταβλητής x_i. Προσοχή, πρέπει να είναι επεκτάσιμες όλες οι συνεπείς πλειάδες του* $R_{S-\{x_i\}}$.

Ας δούμε το αντίστοιχο ορισμό της συνέπειας μονοπατιού, τη σχεσιακή συνέπεια μονοπατιού, (relational path consistency):

Ορισμός: Σχεσιακή συνέπεια μονοπατιού

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Έστω ένας περιορισμός R_S ορισμένος σε ένα σύνολο μεταβλητών S και ένας περιορισμός R_T ορισμένος σε ένα σύνολο μεταβλητών T. Έστω μία τυχαία μεταβλητή* $x_i \in S \cap T$ *και ένας περιορισμός* $R_{S \cup T-\{x_i\}}$ *ορισμένος στο* $S \cup T-\{x_i\}$. *Οι R_S και R_T είναι σχεσιακά συνεπείς ως προς το μονοπάτι σε σχέση με τη x_i (relational path-consistent relative to x_i), εάν και μόνον εάν ο* $R_{S \cup T-\{x_i\}}$ *μπορεί να επεκταθεί στον R_S για κάποια τιμή* $a_{x_i} \in D_{x_i}$ *της μεταβλητής x_i. Προσοχή, πρέπει να είναι επεκτάσιμες όλες οι συνεπείς πλειάδες του* $R_{S \cup T-\{x_i\}}$.

Ας δούμε το αντίστοιχο ορισμό της i-συνέπειας, τη σχεσιακή i-συνέπεια, (relational m-consistency):

Ορισμός: Σχεσιακή m-συνέπεια

> *Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Έστω m το πλήθος σύνολο περιορισμών* $S = R_{S_1}, ..., R_{S_m}$ *ορισμένων στα σύνολα μεταβλητών* $S_1, ..., S_m$ *αντίστοιχα. Έστω μία τυχαία μεταβλητή* $x_j \in \bigcap_{i=1}^{m} S_i$ *, ένα σύνολο* $A = \bigcup_{i=1}^{m} S_i - \{x_j\}$ *και ένας περιορισμός R_A ορισμένος στο A. Το S είναι*

σχεσιακά m-συνεπές σε σχέση με τη x_j (relational m-consistent relative to x_l), εάν και μόνον εάν ο R_A μπορεί να επεκταθεί στον R_S για κάποια τιμή $a_{x_j} \in D_{x_j}$ της μεταβλητής x_j. Προσοχή, πρέπει να είναι επεκτάσιμες όλες οι συνεπείς πλειάδες του R_A.

Ας δούμε το αντίστοιχο ορισμό της κατευθυνόμενης i-συνέπειας, την κατευθυνόμενη σχεσιακή i-συνέπεια, (directional relational m-consistency):

Ορισμός: Κατευθυνόμενη σχεσιακή m-συνέπεια

Έστω ότι έχουμε ένα δίκτυο με περιορισμούς R = (X,D,C). Έστω m το πλήθος τυχαίο σύνολο περιορισμών $S = R_{S_1}, ..., R_{S_m}$ ορισμένων στα σύνολα μεταβλητών $S_1, ..., S_m$ αντίστοιχα και μια απαρίθμηση d των μεταβλητών του R. Έστω x_l μεταβλητή τέτοια που είναι η πιο δεξιά στη d και $x_l \in \{S_1 \cup ... \cup S_m\}$. Έστω ένα τυχαίο σύνολο $A \subseteq \{S_1 \cup ... \cup S_m\} - \{x_l\}$ και ένας περιορισμός R_A ορισμένος στο A. Το R είναι κατευθυνόμενα σχεσιακά m-συνεπές (directionally m-consistent), εάν και μόνον εάν ο R_A μπορεί να επεκταθεί στον R_S για κάποια τιμή $a_{x_j} \in D_{x_j}$ της μεταβλητής x_j. Προσοχή, πρέπει να είναι επεκτάσιμες όλες οι συνεπείς πλειάδες του R_A.

2.3.8.1 Αλγόριθμοι και πολυπλοκότητα για κατευθυνόμενη συνέπεια

Ο αλγόριθμος DRC_m εφαρμόζει κατευθυνόμενη σχεσιακή m-συνέπεια με πολυπλοκότητα $O(nm(2^m k^2)^{w^*(d)+1})$.

2.4 Βιβλιογραφικές αναφορές

Ενδιαφέρον CSP, είναι το πρόβλημα της ζέβρας (the zebra problem). Σε αυτό το πρόβλημα, αναδεικνύεται υπέροχα ο πιο συχνός περιορισμός στα CSP, ο περιορισμός που απαιτεί όλες οι μεταβλητές να μην είναι ίσες (all different constraint) [Rég94]. Ενδιαφέρουσα είναι και η σχέση της αυστηρότητας (tightness) στους περιορισμούς [Zha04].

Ο αλγόριθμος AC-5(R) είναι πολύ γρήγορος για κάποια είδη περιορισμών. Αξίζει η μελέτη του, καθώς τα είδη των περιορισμών που απαιτεί συναντιόνται αρκετά συχνά [DH91].

Ο αλγόριθμος PC-4(R) είναι ο βέλτιστος αλγόριθμος για PC, με πολυπλοκότητα $\Omega(n^3k^3)$ [MH86].

Για i-consistency ο βέλτιστος αλγόριθμος ονομάζεται KS [Coo89] και έχει πολυπλοκότητα $\Omega((nk))^i$.

3 Προβλήματα βελτιστοποίησης με περιορισμούς

3.1 Το συνδυαστικό πρόβλημα της δημοπρασίας

Γιατί στο πρόβλημα της δημοπρασίας θέλουμε να βρούμε όλα τα σύνολα δημοπρασιών; Δεν είναι αρκετό στο δημοπράτη να διαλέξει το πρώτο σύνολο και να ξεκινήσει τη δημοπρασία; Έχουμε σκοπίμως παραλείψει να αναφερθούμε στο ζουμί της υπόθεσης, τα χρήματα. Ο δημοπράτης για κάθε δημοπρασία που δημιουργεί, της αντιστοιχίζει μία τιμή, το κέρδος που θα βγάλει. Σε μία δημοπρασία b_1 που έχει τα προϊόντα a_1, a_2, a_3 και a_4 αντιστοιχίζει μια τιμή r_1. Για μια άλλη δημοπρασία b_2 που συμπεριλαμβάνει τα προϊόντα a_2, a_3 και a_6 έχει αντιστοιχίσει μία άλλη τιμή r_2. Το πιο ενδιαφέρον τώρα πρόβλημα, είναι να βρεθεί αυτό το σύνολο των δημοπρασιών που θα αποφέρει το μεγαλύτερο κέρδος. Το πρόβλημα της δημοπρασίας μετονομάζεται σε συνδυαστικό πρόβλημα της δημοπρασίας και θα αναφερόμαστε πλέον σε αυτό με το ακρωνύμιο CAP από τον αγγλικό όρο combinatorial auction problem.

3.1.1 Στιγμιότυπο του CAP

Έχουμε:

- τα προϊόντα $S = \{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8\}$
- τις δημοπρασίες $B = \{b_1,b_2,b_3,b_4,b_5\}$, όπου
  - $b_1 = (\{a_1,a_2,a_3,a_4\},r_1)$
  - $b_2 = (\{a_2,a_3,a_6\},r_2)$
  - $b_3 = (\{a_1,a_4,a_5\},r_3)$
  - $b_4 = (\{a_2,a_8\},r_4)$
  - $b_5 = (\{a_5,a_6\},r_5)$
- τους περιορισμούς $C = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου
  $R_{12} = R_{13} = R_{14} = R_{24} = R_{25} = R_{35} = \{(1,0),(0,1),(0,0)\}$
- έστω B' οι δημοπρασίες που έχουν επιλεγεί. Τότε θέλουμε την μεγιστοποίηση του:
  $$C(B') = \sum_{b_i \in B'} r_i$$

Το στιγμιότυπο του CAP που περιγράψαμε το ονομάζουμε *P*.

3.2 Δίκτυα με κόστη

Για να περιγράψουμε το CAP, οφείλουμε να ενισχύσουμε λίγο το μοντέλο μας.

Ορισμός: Δίκτυο με κόστος

> *Ορίζουμε σαν δίκτυο με κόστη (cost network) την τετράδα (X,D,C_h,C_s), όπου X είναι το σύνολο των μεταβλητών, D είναι το σύνολο των πεδίων τιμών των μεταβλητών, C_h είναι το σύνολο των ισχυρών περιορισμών και C_s είναι το σύνολο των ασθενών περιορισμών.*
>
> *Ισχυροί (hard constraints) καλούνται οι περιορισμοί C που γνωρίσαμε στα δίκτυα με περιορισμούς και είναι απαραίτητο να ικανοποιηθούν. Για παράδειγμα, στο CAP, ότι πρέπει οπωσδήποτε δύο δημοπρασίες να μην συμπεριλαμβάνουν κοινά προϊόντα.*
>
> *Ασθενείς (soft constraints) στον αντίποδα καλούνται οι περιορισμοί που δεν είναι σημαντικοί, που είναι κάπως αφηρημένοι. Στο CAP, ασθενής είναι ο περιορισμός που θέλει τη μεγιστοποίηση του κέρδους. Ασθενείς μπορούμε να*

καλέσουμε τους περιορισμούς που από τη φύση τους δε θα μας οδηγήσουν σε ασυνέπεια.

Ας εξετάσουμε πιο προσεκτικά τους ασθενείς περιορισμούς. Στα δίκτυα με περιορισμούς είχαμε αναφέρει ότι περιορισμός είναι μία σχέση μεταξύ πεδίων τιμών μεταβλητών. Τι είναι ασθενής περιορισμός; Είναι μια συνάρτηση που έχει σαν πεδίο ορισμού πεδία τιμών μεταβλητών! Στο CAP, ο ασθενής περιορισμός είναι οι συναρτήσεις – τιμές που αναθέτει ο δημοπράτης σε κάθε δημοπρασία (ομάδα από προϊόντα). Οπότε έχουμε C_s *= {το σύνολο των* F_{Qi}*}, όπου* F_{Qi} *είναι συναρτήσεις με πεδίο ορισμού το καρτεσιανό γινόμενο των πεδίων τιμών των μεταβλητών και πεδίο τιμών τους θετικούς πραγματικούς αριθμούς,* F_{qi}*:* $(a_{i1},...,a_{ik}) \rightarrow Reals^{+}$*, με* $a_1 \varepsilon D_1$*, ...,* $a_k \varepsilon D_k$ *και* Q_i *ένα σύνολο μεταβλητών.*

Θέλουμε τη μεγιστοποίηση του κέρδους του CAP. Ορίζουμε τη συνολική συνάρτηση κόστους F (global cost function) ως το άθροισμα όλων των συναρτήσεων των ασθενών περιορισμών, $F = \sum_{j=1}^{k} F_{Q_j}$ *. Στόχος είναι να βρούμε αποτίμηση (assignment)* $\bar{a}^0 = (a_{1,...}, a_n)$ *, έτσι ώστε να μεγιστοποιήσουμε (ή σε άλλα προβλήματα να ελαχιστοποιήσουμε) τη συνολική συνάρτηση κόστους. Αξίζει να σημειώσουμε ότι τη συνολική συνάρτηση κόστους μπορούμε να τη βρούμε στη βιβλιογραφία σαν συνάρτηση κριτήριο (criterion function) ή αντικειμενική συνάρτηση (objective function). Τη συνολική συνάρτηση κόστους θα την καλούμε για συντομία συνάρτηση κόστους ή και απλά κόστος.*

3.2.1 Δίκτυο με κόστη για το CAP

Είναι πλέον πολύ ενδιαφέρον να δούμε τη μοντελοποίηση του CAP με δίκτυο με κόστη. Έχουμε:

- Οι μεταβλητές X είναι οι δημοπρασίες.
- Τα πεδία τιμών D των μεταβλητών είναι πάλι 0 ή 1, όπως και στο δίκτυο με περιορισμούς. 1 για δημοπρασία που τελικά επιλέγεται από τον πλειοδότη, 0 αλλιώς.
- Οι ισχυροί περιορισμοί C_h είναι οι ίδιοι με του δικτύου με περιορισμούς. Εάν δύο δημοπρασίες έχουν κοινά προϊόντα, δεν μπορούν να έχουν επιλεγεί και οι δύο, να έχουν αποτιμηθεί και οι δύο με την τιμή 1.
- Οι ασθενείς περιορισμοί C_s είναι το νέο και πιο ενδιαφέρον κομμάτι. Οι συναρτήσεις αυτές έχουν σαν πεδίο ορισμού το πεδίο τιμών μίας μόνο μεταβλητής (δημοπρασίας) και σαν πεδίο τιμών το κέρδος που θα αποφέρει αυτή.
- Η συνάρτηση κόστους F είναι επίσης ενδιαφέρουσα. Είναι το άθροισμα όλων των ασθενών περιορισμών (των κερδών των δημοπρασιών). Ζητούμε τη μεγιστοποίηση αυτής της συνάρτησης.

3.2.2 Δίκτυο με κόστη για στιγμιότυπο του CAP

Ας δούμε πιο αναλυτικά ένα δίκτυο με κόστος για το στιγμιότυπο *P* του CAP. Παρόμοια με το δίκτυο με περιορισμούς έχουμε:

- $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12} = R_{13} = R_{14} = R_{24} = R_{25} = R_{35} = \{(1,0),(0,1),(0,0)\}$
- $X = \{x_1,x_2,x_3,x_4,x_5\}$
- $D = \{D_1,D_2,D_3,D_4,D_5\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = \{0,1\}$
- $Q_i = \{x_i\}$, για i=[1,5]

- ➔ $C_s = \{F_1,F_2,F_3,F_4,F_5\}$, με $F_i:D_i \rightarrow Reals^+$ και $F_i(x_i)=\begin{cases} r_i, \text{if } x_i=1 \\ 0, \text{αλλιώς} \end{cases}$ και i=[1,5]
- ➔ $F=\sum_{i=1}^{5} F_i$
- ➔ Θέλουμε: $\max_{x_1,x_2,x_3,x_4,x_5} F$

3.2.3 Δίκτυο με κόστη για CAP με αριθμητικά δεδομένα

Για την πλήρη κατανόηση της μοντελοποίησης παραθέτουμε ένα παράδειγμα CAP με αριθμητικά δεδομένα.

- ➔ Έστω ότι έχουμε οχτώ προϊόντα $\{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8\}$ που θέλουμε να πουληθούν. Ο δημοπράτης φτιάχνει τις εξής δημοπρασίες :
  - $b_1 = (\{a_1,a_2,a_3,a_4\},8)$
  - $b_2 = (\{a_2,a_3,a_6\},6)$
  - $b_3 = (\{a_1,a_4,a_5\},5)$
  - $b_4 = (\{a_2,a_8\},2)$
  - $b_5 = (\{a_5,a_6\},2)$
- ➔ Για τη b_i δημοπρασία ο δημοπράτης θα εισπράξει 8, κτλ. Περιγράφουμε τους ασθενείς περιορισμούς! Έχουμε:
  - $F_1(x_1)=\begin{cases} 8, \text{εάν } x_1=1 \\ 0, \text{αλλιώς} \end{cases}$
  - $F_2(x_2)=\begin{cases} 6, \text{εάν } x_2=1 \\ 0, \text{αλλιώς} \end{cases}$
  - $F_3(x_3)=\begin{cases} 5, \text{εάν } x_3=1 \\ 0, \text{αλλιώς} \end{cases}$
  - $F_4(x_4)=\begin{cases} 2, \text{εάν } x_4=1 \\ 0, \text{αλλιώς} \end{cases}$
  - $F_5(x_5)=\begin{cases} 2, \text{εάν } x_5=1 \\ 0, \text{αλλιώς} \end{cases}$

  Πρέπει να μεγιστοποιήσουμε τη συνάρτηση $F=\sum_{i=1}^{5} F_i$, θέλουμε $\max_{x_1,x_2,x_3,x_4,x_5} F$.
- ➔ Οι μεταβλητές είναι $x_i = b_i$ για $i \in \{1,2,3,4,5\}$
- ➔ Τα πεδία τιμών των μεταβλητών είναι 0 ή 1, 1 εάν η δημοπρασία τελικά επιλέγεται από το δημοπράτη, 0 σε άλλη περίπτωση.
- ➔ Οι ισχυροί περιορισμοί είναι ότι δύο δημοπρασίες δεν μπορούν να έχουν κοινά προϊόντα: $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}= R_{13}=R_{14}=R_{24}=R_{25}=R_{35}=\{(1,0),(0,1),(0,0)\}$.
- ➔ Για να λύσουμε το πρόβλημα οφείλουμε να ικανοποιήσουμε τους ισχυρούς και ασθενείς περιορισμούς.

3.3 Μέθοδος επίλυσης προβλημάτων με περιορισμούς

Ας θυμηθούμε το AP. Σε αυτό, δε παίρναμε υπόψιν τα κέρδη του δημοπράτη και δεν είχαμε ασθενείς περιορισμούς. Τέτοια προβλήματα, με μόνο ισχυρούς περιορισμούς, τα καλούμε «προβλήματα με περιορισμούς (constraint problems)». Χάριν συντομίας, τα προβλήματα με περιορισμούς θα τα καλούμε CP από το ακρωνύμιο που σχηματίζεται από τους αγγλικούς όρους constraint problems.

Εάν υπολογίσουμε και τα κέρδη του δημοπράτη, έχουμε το CAP. Το CAP ανήκει σε μια «οικογένεια» προβλημάτων που ονομάζεται «προβλήματα βελτιστοποίησης με περιορισμούς

(constraint optimization problems)». Χάριν συντομίας τα προβλήματα βελτιστοποίησης με περιορισμούς θα τα καλούμε COP από το ακρωνύμιο που σχηματίζεται από τους αγγλικούς όρους contraint optimization problems. Η διαδικασία επίλυσης τέτοιων προβλημάτων λέγεται «βελτιστοποίηση με περιορισμούς (constraint optimization)». Όλες αυτές οι διαδικασίες επίλυσης μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες, στους τρόπους επίλυσης με αναζήτηση και σε αυτούς με συμπεράσματα. Έτσι, οι αλγόριθμοι, που επιλύουν τέτοια προβλήματα, μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες: στους αλγόριθμους με αναζήτηση (search) και στους αλγόριθμους με συμπερασμούς (inference). Ωστόσο, οι πιο αποδοτικοί αλγόριθμοι, που χρησιμοποιούνται και στην πράξη, είναι αυτοί που συνδυάζουν αναζήτηση και συμπερασμό.

Είναι φανερό ότι όλοι οι αλγόριθμοι που υπάρχουν για CP μπορούν να επεκταθούν για COP. Στην εργασία αυτή θα ασχοληθούμε με αλγόριθμους συμπερασμού για COP. Συγκεκριμένα, θα ασχοληθούμε με μια οικογένεια τέτοιων αλγορίθμων που ονομάζονται «αλγόριθμοι απαλοιφής με κάδους (bucket elimination algorithms)». Πάλι για λόγους συντομίας, θα καλούμε BE τους αλγόριθμους απαλοιφής με κάδους από το ακρωνύμιο που σχηματίζεται από τους αγγλικούς όρους bucket elimination.

Οι BE είναι δυναμικός προγραμματισμός (dynamic programming) και ουσιαστικά είναι το ανάλογο του αλγορίθμου «προσαρμοζόμενη συνέπεια» (adaptive consistency) για CP.

3.4 Απαλοιφή μεταβλητών

Ένα άλλο όνομα για τους BE είναι «αλγόριθμοι απαλοιφής μεταβλητών (variable elimination algorithms)». Ίσως αυτό το όνομα είναι και το πιο «σωστό» από τεχνικής άποψης, καθώς αντικατοπτρίζει ακριβώς το τι κάνουν οι BE. Σε κάθε βήμα αυτών των αλγορίθμων αφαιρείται και από μία μεταβλητή του προβλήματος. Για λόγους συντομίας θα αναφερόμαστε στην απαλοιφή μεταβλητών με το ακρωνύμιο VE από τον αγγλικό όρο variable elimination.

3.4.1 «Ενισχυμένο» στιγμιότυπο του CAP, με ασθενείς περιορισμούς που έχουν arity > 1

Για να καταλάβουμε καλύτερα το τι γίνεται με τη VE, θα την κάνουμε για ένα παράδειγμα του CAP με αριθμητικά δεδομένα. Δε θα χρησιμοποιήσουμε ακριβώς το παράδειγμα του CAP που έχουμε δώσει, αλλά ένα άλλο πιο ενισχυμένο, όπου θα έχουμε και συναρτήσεις – ασθενείς περιορισμούς με πεδίο ορισμού σύνολο πεδίων τιμών μεταβλητών πληθικά μεγαλύτερο από μονοσύνολο.

Έστω ότι ο δημοπράτης θέλει να δημοπρατήσει για κάποιο λόγο κάποια προϊόντα μαζί. Εάν πουλήσει το a_1, καλό θα ήταν να πουλήσει και το a_2 γιατί με αυτόν τον τρόπο γλιτώνει κάποια έξοδα. Οπότε έχουμε ασθενείς περιορισμούς – κέρδος σε πάνω από μία μεταβλητές ταυτόχρονα και ορίστε το παράδειγμά μας:

- Έχουμε οχτώ προϊόντα $\{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8\}$. Οι δημοπρασίες και αυτές είναι :
  - $b_1 = (\{a_1,a_2,a_3,a_4\},8)$
  - $b_2 = (\{a_2,a_3,a_6\},6)$
  - $b_3 = (\{a_1,a_4,a_5\},5)$
  - $b_4 = (\{a_2,a_8\},2)$
  - $b_5 = (\{a_5,a_6\},2)$
  - $b_6 = (\{a_7\},10)$
- Έχουμε:
  - $F_1(x_1)=\begin{cases}8, \text{εάν } x_1=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_2(x_2)=\begin{cases}6, \text{εάν } x_2=1\\ 0, \text{αλλιώς}\end{cases}$

- $F_3(x_3)=\begin{Bmatrix}5,\text{εάν } x_3=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_4(x_4)=\begin{Bmatrix}2,\text{εάν } x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_5(x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_6(x_6)=\begin{Bmatrix}4,\text{εάν } x_6=1\\0,\text{αλλιώς}\end{Bmatrix}$

Επιπλέον έχουμε τις εξής συναρτήσεις – ασθενείς περιορισμούς:

- $F_{12}(x_1,x_2)=\begin{Bmatrix}10,\text{εάν } x_1=x_2=1\\8,\text{εάν } x_1=x_2=0\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{13}(x_1,x_3)=\begin{Bmatrix}8,\text{εάν } x_1=x_3=1\\2,\text{εάν } x_1=1 \text{ και } x_3=0\\10,\text{εάν } x_1=0 \text{ και } x_3=0\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix}8,\text{εάν } x_2=x_3=x_5\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix}8,\text{εάν } x_1=x_2=x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{56}(x_5,x_6)=\begin{Bmatrix}10,\text{εάν } x_5=x_6=1\\6,\text{εάν } x_5=x_6=0\\0,\text{αλλιώς}\end{Bmatrix}$

➔ Θέλουμε, $\max_{x_1,x_2,x_3,x_4,x_5} \sum F_i$.

➔ Οι μεταβλητές είναι $x_i = b_i$ για i=[1,6].

➔ Τα πεδία τιμών των μεταβλητών είναι 0 ή 1.

➔ $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}= R_{13} =R_{14} =R_{24} =R_{25} =R_{35} =\{(1,0),(0,1),(0,0)\}$.

Θέλουμε να μεγιστοποιήσουμε τη συνάρτηση κόστους, F. Ας ονομάσουμε M τη μέγιστη τιμή αυτής της συνάρτησης.

3.4.2 Παράδειγμα VE

Ας παίξουμε λίγο με πράξεις.

$$M=\max_{x_1,x_2,x_3,x_4,x_5,x_6} F=\max_{x_1,x_2,x_3,x_4,x_5,x_6}\sum F_i=$$

$$=\max_{x_1,x_2,x_3,x_4,x_5,x_6}\{F_1(x_1)+F_2(x_2)+F_3(x_3)+F_4(x_4)+F_5(x_5)+F_6(x_6)+$$

$$+F_{12}(x_1,x_2)+F_{13}(x_1,x_3)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+F_{56}(x_5,x_6)\}$$

Οι ΒΕ απαλείφουν μία μεταβλητή κατά μία απαρίθμηση (order) των μεταβλητών. Η επιλογή της απαρίθμησης είναι κρίσιμη για την πολυπλοκότητα των ΒΕ. Για διαφορετικές απαριθμήσεις μπορεί να έχουμε μεγάλη διαφορά στην πολυπλοκότητα και ως προς το χρόνο και ως προς το χώρο. Προσωρινά, δεν θα ασχοληθούμε με την επιλογή της απαρίθμησης και αργότερα θα επανέλθουμε αποδεικνύοντας τη σοβαρότητά της.

Για το παράδειγμά μας, επιλέγουμε εντελώς τυχαία την απαρίθμηση $d_1 = (x_1,x_3,x_2,x_5,x_4,x_6)$.

Πρώτα απαλείφουμε την τελευταία μεταβλητή στην απαρίθμησή μας, την x_6.

$$M = \max_{x_1,x_2,x_3,x_4,x_5} \{F_1(x_1)+F_2(x_2)+F_3(x_3)+F_4(x_4)+F_5(x_5)+F_{12}(x_1,x_2)+$$

$$+F_{13}(x_1,x_3)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}$$

Η VE είναι κατά κάποιο τρόπο η ανεξαρτητοποίησή της μεταβλητής από τον υπολογισμό του μέγιστου (max). Σε λίγο θα φανεί πιο καθαρά.

Ομοίως, για τη x_4 έχουμε:

$$M = \max_{x_1,x_2,x_3,x_5} \{F_1(x_1)+F_2(x_2)+F_3(x_3)+F_5(x_5)+F_{12}(x_1,x_2)+$$

$$+F_{13}(x_1,x_3)+F_{235}(x_2,x_3,x_5)+\max_{x_4}\{F_4(x_4)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}$$

Για τη x_5:

$$M = \max_{x_1,x_2,x_3} \{F_1(x_1)+F_2(x_2)+F_3(x_3)+F_{12}(x_1,x_2)+F_{13}(x_1,x_3)+$$

$$+\max_{x_5}\{F_5(x_5)+F_{235}(x_2,x_3,x_5)+\max_{x_4}\{F_4(x_4)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}\}$$

Για τη x_2:

$$M = \max_{x_1,x_3} \{F_1(x_1)+F_3(x_3)+F_{13}(x_1,x_3)+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+$$

$$+\max_{x_5}\{F_5(x_5)+F_{235}(x_2,x_3,x_5)+\max_{x_4}\{F_4(x_4)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}\}$$

Για τη x_3.

$$M = \max_{x_1} \{F_1(x_1)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+$$

$$+\max_{x_5}\{F_5(x_5)+F_{235}(x_2,x_3,x_5)+\max_{x_4}\{F_4(x_4)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}\}\}$$

Επιτέλους, θα ξεκινήσουμε τις απαλοιφές των μεταβλητών. Είναι σχεδόν φανερό το πως θα γίνει η απαλοιφή. Ο όρος $\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}$ είναι ο μοναδικός που περιέχει x_6. Μπορούμε να τον υπολογίσουμε μόνο ως προς τη x_6 και να πάρουμε μία νέα συνάρτηση που δεν την έχει στο πεδίο ορισμού της, $h^{x_6}(x_5)=\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}$.

Έχουμε:

$$M = \max_{x_1} \{F_1(x_1)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+$$

$$+\max_{x_5}\{F_5(x_5)+F_{235}(x_2,x_3,x_5)+\max_{x_4}\{F_4(x_4)+F_{124}(x_1,x_2,x_4)+h^{x_6}(x_5)\}\}\}\}$$

Βλέπουμε ότι πλέον έχουμε απαλείψει τη μεταβλητή x_6! Είναι σαν να έχουμε μία μεταβλητή λιγότερη. Θα συνεχίσουμε απαλείφοντας τη x_4 ορίζοντας τη συνάρτηση:

$$h^{x_4}(x_1,x_2)=\max_{x_4}\{F_4(x_4)+F_{124}(x_1,x_2,x_4)\}$$

Θα πάρουμε για M μία συνάρτηση με λιγότερα ορίσματα:

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+$$

$$+\max_{x_5}\{F_5(x_5)+F_{235}(x_2,x_3,x_5)+h^{x_4}(x_1,x_2)+h^{x_6}(x_5)\}\}\}\}$$

Για τη μεταβλητή x_5 ορίζουμε:

$$h^{x_5}(x_2,x_3)=\max_{x_5}\{F_5(x_5)+F_{235}(x_2,x_3,x_5)+h^{x_6}(x_5)\}$$

και παίρνουμε:

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+h^{x_5}(x_2,x_3)+$$

$$+h^{x_4}(x_1,x_2)\}\}\}$$

Για τη μεταβλητή x_2 ορίζουμε:

$$h^{x_2}(x_1,x_3)=\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+h^{x_5}(x_2,x_3)+h^{x_4}(x_1,x_2)\}$$

και παίρνουμε:

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+h^{x_2}(x_1,x_3)\}\}$$

Για τη μεταβλητή x_3 ορίζουμε:

$$h^{x_3}(x_1)=\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+h^{x_2}(x_1,x_3)\}$$

και παίρνουμε:

$$M=\max_{x_1}\{F_1(x_1)+h^{x_3}(x_1)\}$$

Τέλος, για τη μεταβλητή x_1 ορίζουμε:

$$h^{x_1}=\max_{x_1}\{F_1(x_1)+h^{x_3}(x_1)\}$$

και παίρνουμε:

$$M=h^{x_1}$$

Δε θα υπολογίσουμε το Μ για το παράδειγμά μας. Θα το κάνουμε στο επόμενο κεφάλαιο μέσω ενός αλγορίθμου που υλοποιεί την παραπάνω διαδικασία.

4 Απαλοιφή με κάδους

4.1 Γιατί κάδοι;

Έχουμε αναφέρει ότι τη VE θα τη βρούμε σαν BE. Οι BE (αλγόριθμοι απαλοιφής με κάδους) βλέπουν την όλη διαδικασία από μία διαφορετική αλλά ισοδύναμη σκοπιά, από αυτή των κάδων. Σε κάθε μεταβλητή αναθέτουν και από ένα κάδο. Στην πραγματικότητα, όπως θα δούμε, οι BE είναι η υλοποίηση της VE, όπως ακριβώς ένα πρόγραμμα είναι υλοποίηση ενός αλγορίθμου. Το παράξενο είναι ότι στη βιβλιογραφία η ορολογία BE έχει επικρατήσει της VE.

4.1.1 Παράδειγμα BE

Για ευκολία, παραθέτουμε εν συντομία το ενισχυμένο παράδειγμα CAP εδώ.

- $X = \{x_1,x_2,x_3,x_4,x_5,x_6\}$, όπου το x_i συμβολίζει τη δημοπρασία b_i
- $D = \{D_1,D_2,D_3,D_4,D_5,D_6\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = \{0,1\}$.
- Ασθενείς περιορισμοί C_s:
  - $F_1(x_1)=\begin{Bmatrix}8,\text{εάν } x_1=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_2(x_2)=\begin{Bmatrix}6,\text{εάν } x_2=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_3(x_3)=\begin{Bmatrix}5,\text{εάν } x_3=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_4(x_4)=\begin{Bmatrix}2,\text{εάν } x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_5(x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_6(x_6)=\begin{Bmatrix}4,\text{εάν } x_6=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{12}(x_1,x_2)=\begin{Bmatrix}10,\text{εάν } x_1=x_2=1\\8,\text{εάν } x_1=x_2=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{13}(x_1,x_3)=\begin{Bmatrix}8,\text{εάν } x_1=x_3=1\\2,\text{εάν } x_1=1 \text{ και } x_3=0\\10,\text{εάν } x_1=0 \text{ και } x_3=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix}8,\text{εάν } x_2=x_3=x_5\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix}8,\text{εάν } x_1=x_2=x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{56}(x_5,x_6)=\begin{Bmatrix}10,\text{εάν } x_5=x_6=1\\6,\text{εάν } x_5=x_6=0\\0,\text{αλλιώς}\end{Bmatrix}$
- Ισχυροί περιορισμοί, $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}=R_{13}=R_{14}=R_{24}=R_{25}=R_{35}=\{(1,0),(0,1),(0,0)\}$.

Ας δούμε καλύτερα στο παράδειγμά μας ποιοι είναι οι κάδοι. Έχουμε τους Bucket(x_1),

Bucket(x_2), Bucket(x_3), Bucket(x_4), Bucket(x_5) και Bucket(x_6). Ας δούμε για την απαρίθμηση d_1 = (x_1,x_3,x_2,x_5,x_4,x_6) τι γίνεται ακριβώς.

- ➔ Όπως και στη VE, παίρνουμε τη τελευταία μεταβλητή της απαρίθμησης, τη x_6. Βλέπουμε ποιες συναρτήσεις – ασθενείς περιορισμοί έχουν σαν όρισμα τη μεταβλητή αυτή, η $F_6(x_6)$ και η $F_{56}(x_5,x_6)$, και τις τοποθετούμε στον κάδο Bucket(x_6).
- ➔ Μετά παίρνουμε τη x_4 και στον κάδο Bucket(x_4) τοποθετούμε τις συναρτήσεις που έχουν όρισμα τη x_4 και δεν έχουν τοποθετηθεί σε κάποιον κάδο, τη $F_4(x_4)$ και τη $F_{124}(x_1,x_2,x_4)$.
- ➔ Για τη x_5 τοποθετούμε στο Bucket(x_5) τις $F_5(x_5)$ και τη $F_{235}(x_1,x_3)$.
- ➔ Για τη x_2 στο Bucket(x_2) τοποθετούμε τη $F_2(x_2)$ και τη $F_{12}(x_1,x_2)$. Τις $F_{235}(x_2,x_3,x_5)$ και $F_{124}(x_1,x_2,x_4)$ δε τις τοποθετούμε γιατί έχουν τοποθετηθεί σε άλλους κάδους. Η $F_{124}(x_1,x_2,x_4)$ στο Bucket(x_4) και η $F_{235}(x_2,x_3,x_5)$ στο Bucket(x_5).
- ➔ Για τη x_3 στο Bucket(x_3) τοποθετούμε τη $F_3(x_3)$ και τη $F_{13}(x_1,x_3)$.
- ➔ Για τη x_1 στο Bucket(x_1) τοποθετούμε τη $F_1(x_1)$.

Έχουμε:

Κάδος	Συναρτήσεις
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5,x_6)$
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2,x_3,x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1,x_3)$
Bucket(x_1):	$F_1(x_1)$

Πίνακας 1: Οι κάδοι αρχικά

Πώς θα υπολογίσουμε τη συνάρτηση κέρδους M; Πώς θα χρησιμοποιήσουμε τους κάδους; Σε κάθε κάδο υπάρχει ένα μικρότερο πρόβλημα CAP που πρέπει να λυθεί ως προς μία μεταβλητή, αυτή του κάδου. Μετά το μικρότερο λυμένο αυτό πρόβλημα θα το ρίξουμε στον επόμενο κάδο και θα λύσουμε ξανά ένα άλλο καινούριο CAP και ούτω καθεξής μέχρι να μας τελειώσουν οι κάδοι – μεταβλητές. Θα μπορούσαμε να καλέσουμε τον αλγόριθμο αυτό ”απαλοιφή κάδων”.

Ας δούμε αναλυτικά τι γίνεται μέσα στους κάδους. Στην αρχή υπολογίζουμε τη $h^{x_6}(x_5)$ και την τοποθετούμε στον κάδο Bucket(x_5). Μετά υπολογίζουμε τη $h^{x_4}(x_1,x_2)$. Πού θα τοποθετηθεί η $h^{x_4}(x_1,x_2)$; Στον κάδο Bucket(x_1) ή στο Bucket(x_2); Θα τοποθετηθεί στο bucket(x_2) γιατί η x_2 είναι πιο δεξιά από τη x_1 στην d_1.

Για να δείτε γιατί πρέπει να είναι πιο δεξιά σκεφτείτε το εξής: Έστω ότι τη $h^{x_4}(x_1,\chi_2)$, την τοποθετούμε στο Bucket(x_1). Όταν θα τελειώναμε με τους υπολογισμούς μας στο Bucket(x_2) θα είχαμε την εντύπωση ότι θα είχαμε απαλείψει τη μεταβλητή x_2, το οποίο όμως δε θα ίσχυε καθώς θα υπήρχε η $h^{x_4}(x_1,x_2)$ στο Bucket(x_1). Κάθε φορά πρέπει να τοποθετούμε τις νέες συναρτήσεις στον κάδο του οποίου η μεταβλητή είναι η πιο δεξιά στην απαρίθμηση που χρησιμοποιούμε.

Τώρα έχουμε:

Κάδος	Συναρτήσεις	
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5,x_6)$	
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$	
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2,x_3,x_5)$,	$h^{x_6}(x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$,	$h^{x_4}(x_1,\chi_2)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1,x_3)$	
Bucket(x_1):	$F_1(x_1)$	

Πίνακας 2: Οι κάδοι μετά την επεξεργασία των Bucket(x_6) και Bucket(x_4)

Είμαστε πλέον στον κάδο Bucket(x_5). Υπολογίζουμε τη $h^{x_5}(x_2,x_3)$. Αυτή θα τοποθετηθεί στον κάδο Bucket(x_2). Μετά θα ασχοληθούμε με τον κάδο Bucket(x_2) και θα υπολογίσουμε τη

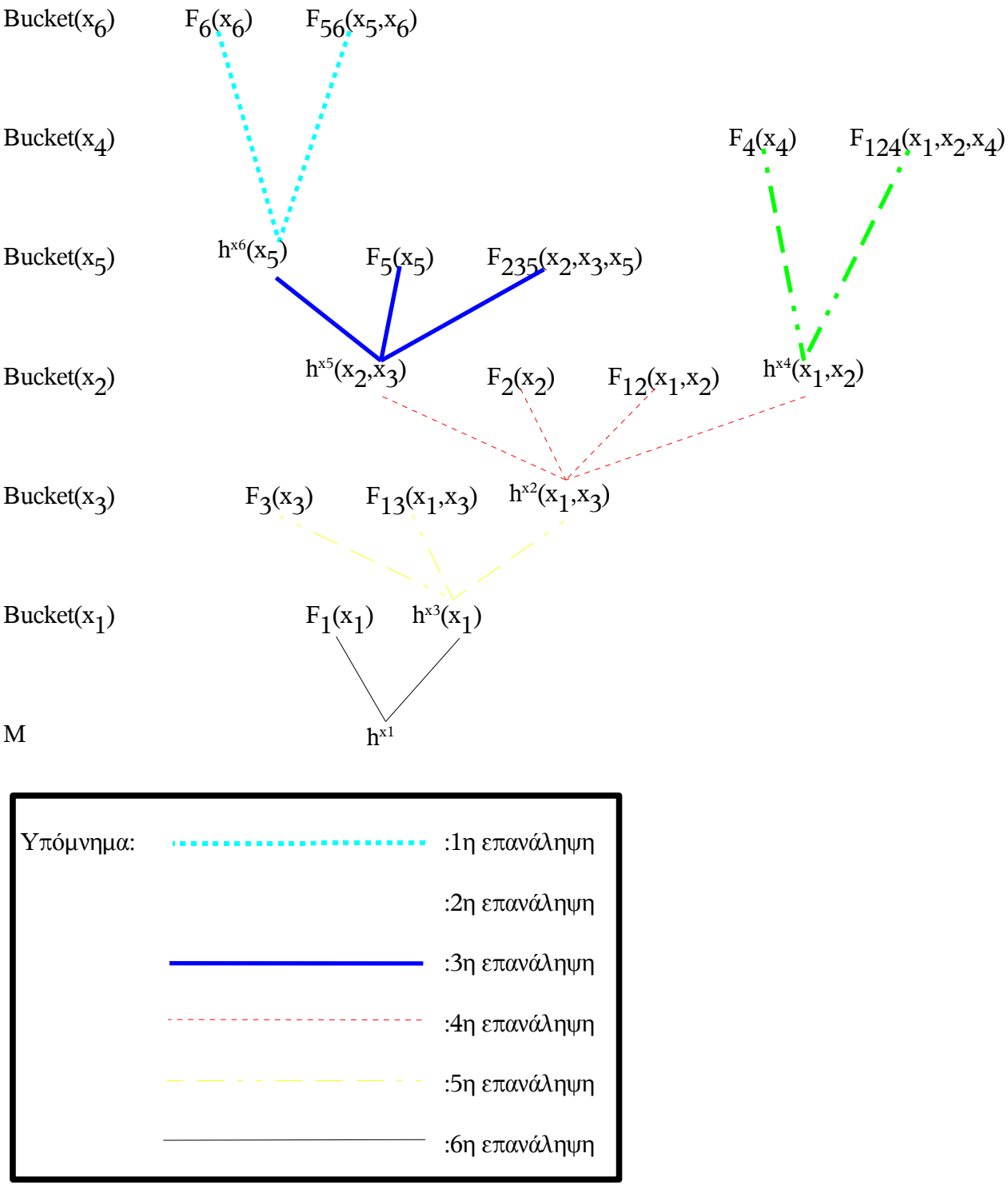
$h^{x_2}(x_1, x_3)$. Έπειτα στο Bucket(x_3) θα υπολογίσουμε τη $h^{x_3}(x_1)$ και θα τη βάλουμε στον κάδο Bucket(x_1), στον οποίο κάδο τελικά θα υπολογίσουμε τη M.

Συνολικά έχουμε:

Κάδος	Συναρτήσεις
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5,x_6)$
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2,x_3,x_5)$, $h^{x_6}(x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $h^{x_4}(x_1,\chi_2)$, $h^{x_5}(x_2,x_3)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1,x_3)$, $h^{x_2}(x_1,x_3)$
Bucket(x_1):	$F_1(x_1)$, $h^{x_3}(x_1)$

Πίνακας 3: Οι κάδοι μετά την επεξεργασία των Bucket(x_6) και Bucket(x_4)

Παρατηρούμε ότι συνολικά θα λύσουμε το ίδιο πρόβλημα 6 φορές, όσοι είναι και οι κάδοι. Σχηματικά η όλη διαδικασία φαίνεται πιο ωραία:



Σχήμα 2: BE για το ενισχυμένο στιγμιότυπο CAP

4.2 Αλγόριθμος ELIM-OPT

Ωραία όλα αυτά, αλλά πρέπει να παραθέσουμε και έναν πιο ακριβή BE αλγόριθμο, ώστε να μπορέσουμε να μελετήσουμε την πολυπλοκότητά του. Αυτό θα κάνουμε περίπου, μόνο που αρχικά για να γίνει πιο κατανοητός ο αλγόριθμος ,θα παραλείψουμε τους ισχυρούς περιορισμούς. Ορίστε ο πολυπόθητος αλγόριθμος, ο ELIM-OPT:

Αλγόριθμος ELIM-OPT

Δεδομένα:

Ένα δίκτυο με περιορισμούς (X,D,C_s), με $C_s = \{F_1,...,F_r\}$ και $r = |C_s|$.

Μία απαρίθμηση d των μεταβλητών X.

Αρχή Αλγορίθμου:

Για κάθε ασθενή περιορισμό F_t, με t από το 1 στο r, **κάνε:**

Q_t = η εμβέλεια της F_t

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη Q_t

Τοποθέτησε τη F_t στον κάδο Bucket(x_b)

Τέλος Για κάθε

Για κάθε κάδο Bucket(x_p), με p από το n στο 1, **κάνε:**

$h^{x_i},...,h^{x_j}$ = οι συναρτήσεις-περιορισμοί που έχουν προστεθεί στο Bucket(x_p)

$Q_i,...,Q_j$ = οι εμβέλειες των $h^{x_i},...,h^{x_j}$

$F_k,...,F_l$ = οι αρχικοί ασθενείς περιορισμοί που υπάρχουν στο Bucket(x_p)

$Q_k,...,Q_l$ = οι εμβέλειες των $F_k,...,F_l$

$A = (\bigcup_t Q_t) - \{x_p\}$

$$h^{x_p} = \max_{x_p} \{\sum_{t=i}^{j} h^{x_t} + \sum_{t=k}^{l} F_t\}$$

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη A

Τοποθέτησε τη h^{x_p} στο Bucket(x_b)

Τέλος Για κάθε

Για i από το 1 στο n, **κάνε**

Δεδομένης αποτίμησης $\vec{a_{i-1}}$, αποτίμησε το x_i με μια τιμή a_i ε D_i, που μεγιστοποιεί το άθροισμα των συναρτήσεων στο Bucket(x_i)

Τέλος Για

Τέλος Αλγορίθμου

Έξοδος:

Το βέλτιστο κόστος που υπάρχει στο Bucket(x_1).

Η αποτίμηση που απέφερε το βέλτιστο κόστος.

Πριν συνεχίσουμε με την ανάλυση του ELIM-OPT θα παραθέσουμε δύο ορισμούς:

Ορισμός: Ορθός αλγόριθμος

Ένας αλγόριθμος είναι έγκυρος[2] (sound), εάν κάθε αποτέλεσμα που επιστρέφει είναι σωστή λύση.

Ορισμός: Πλήρης αλγόριθμος

Ένας αλγόριθμος είναι πλήρης (complete), εάν επιστρέφει όλες τις πιθανές λύσεις.

Η διαίσθησή μας είναι ότι ο ELIM-OPT επιλύει ένα COP και βρίσκει πάντα μία έγκυρη (valid) αποτίμηση και τη βέλτιστη συνάρτηση κόστους. Η διαίσθηση μας είναι σωστή λόγω του παρακάτω θεωρήματος:

2 Στη βιβλιογραφία, το sound εκτός από έγκυρος μεταφράζεται και σαν ορθός

Θεώρημα: Πληρότητα και ορθότητα ELIM-OPT

Ο αλγόριθμος ELIM-OPT είναι πλήρης και ορθός.

Απόδειξη:

Πρώτα θα δείξουμε ότι ο αλγόριθμος είναι ορθός, δηλαδή ότι έχει σαν αποτέλεσμα τη βέλτιστη συνάρτηση κόστους. Το αποδεικνύουμε με επαγωγή στο πλήθος των μεταβλητών. Χάριν συντομίας, θα καλούμε τη συνάρτηση κόστους, κόστος.

Επαγωγική Βάση:

Για COP, με μόνο μία μεταβλητή, είναι προφανές ότι ο ELIM-OPT υπολογίζει το βέλτιστο κόστος.

Επαγωγική Υπόθεση:

Ο ELIM-OPT βρίσκει το βέλτιστο κόστος για n μεταβλητές.

Επαγωγικό Βήμα:

Έστω ότι έχουμε εκτελέσουμε τον ELIM-OPT σε n μεταβλητές με μια απαρίθμηση d. Θα εκτελέσουμε τον ELIM-OPT για n+1 μεταβλητές με μια απαρίθμηση d'. Η d' προκύπτει από τη d, εάν προσθέσουμε στο τέλος της d τη μεταβλητή x_{n+1}. Ο ELIM-OPT θα ξεκινήσει από τη μεταβλητή x_{n+1}, θα υπολογίσει ένα κόστος $h^{x_{n+1}}$ και θα τοποθετήσει αυτό το κόστος σε ένα άλλο κάδο. Αλλά από την επαγωγική υπόθεση η ELIM-OPT θα υπολογίσει το βέλτιστο κόστος για τους επόμενους n κάδους.

Μένει να δείξουμε ότι ο ELIM-OPT είναι πλήρης, δηλαδή, ότι πάντα, όσες φορές και να τον εκτελέσουμε, θα τερματίσει και θα βρει το βέλτιστο κόστος. Το ότι ο αλγόριθμος τερματίζει είναι πασιφανές. Αλλά αφού ο αλγόριθμος τερματίζει και είναι ορθός, πάντα βρίσκει τη βέλτιστη λύση.

4.2.1 Πολυπλοκότητα ELIM-OPT

Ας μελετήσουμε την πολυπλοκότητα του ELIM-OPT. Αλλά πριν ξεκινήσουμε το σαφάρι στη ζούγκλα με τα O, ας δώσουμε ένα χρήσιμο ορισμό που θα συναντήσει κάποιος πολύ συχνά σε επεξεργασία περιορισμών (constraint processing).

Θυμηθείτε ότι μια συνάρτηση είναι μία σχέση μεταξύ των πεδίων τιμών των μεταβλητών της εμβέλειάς της.

Ορισμός: Πλειάδα

Πλειάδα (tuple) σε μια σχέση με n μεταβλητές, καλούμε τη διατεταγμένη n-άδα που έχει σαν μέλη, τιμές από τα πεδία τιμών των μεταβλητών. Στο i-οστό μέλος έχει ανατίθεται τιμή από το πεδίο τιμών της i-οστής μεταβλητής x_n.

Έχουμε αναφέρει ότι μεγάλο παράγοντα στην πολυπλοκότητα του ELIM-OPT έχει το επαγόμενο πλάτος του διατεταγμένου γράφου με περιορισμούς. Αυτό επιβεβαιώνεται και από το επόμενο θεώρημα:

Θεώρημα: Πολυπλοκότητα ELIM-OPT

Η πολυπλοκότητά σε χώρο και χρόνο του ELIM-OPT για μια απαρίθμηση d, είναι $O(r \cdot k^{w^*(d)})$ *, όπου k είναι η πληθικότητα της μεταβλητής με το μεγαλύτερο σε πληθικότητα πεδίο τιμών και r είναι το πλήθος των ασθενών περιορισμών.*

Απόδειξη:

Πρώτα, θα ασχοληθούμε με την πολυπλοκότητά σε χρόνο. Στο τέλος της ανάλυσης, θα δείξουμε ότι η πολυπλοκότητά σε χώρο είναι η ίδια με αυτή σε χρόνο.

Στην αρχικοποίηση των κάδων δε χρειαζόμαστε πολύ χρόνο, απλά μοιράζουμε r συναρτήσεις σε n κάδους, το οποίο γίνεται σε O(r) βήματα.

Έπειτα, έχουμε για κάθε κάδο Bucket(x_i) να υπολογίσουμε μια νέα συνάρτηση h^{x_i} *και να την τοποθετήσουμε σε ένα άλλο κάδο. Ο υπολογισμός της* h^{x_i} *συνεπάγεται την πρόσθεση* r_i *συναρτήσεων που προϋπάρχουν στον κάδο. Κάθε μία τέτοια συνάρτηση μπορεί να έχει το πολύ* k^{arity} *πλειάδες, όπου με τον όρο arity συμβολίζουμε το πλήθος των ορισμάτων της συνάρτησης, της εμβέλειας. Το πλήθος των πλειάδων μπορεί να είναι ίσο με* k^{arity}*. Έχουμε για το Bucket(x_i), πολυπλοκότητά* $O(r_i \cdot k^{arity})$*. Αυτό το arity όμως είναι λίγο θολό. Πόσο μεγάλο μπορεί να είναι; Το έχουμε απαντήσει. Στο Bucket(x_i) υπάρχουν το πολύ* $w^*(d) + 1$ *μεταβλητές! Συν ένα γιατί υπολογίζουμε και την* x_i*. Άρα στο Bucket(x_i) έχουμε* $O(r_i \cdot k^{w^*(d)+1}) = O(r_i \cdot k^{w^*(d)})$ *. Συνολικά, για τους n κάδους έχουμε* $\sum_{i=1}^{n} O(r_i \cdot k^{w^*(d)+1}) = O(\sum_{i=1}^{n} r_i \cdot k^{w^*(d)}) = O(r \cdot k^{w^*(d)})$ *.*

Στο τέλος του αλγορίθμου, που αποτιμούμε τις μεταβλητές, η αποτίμηση μπορεί να γίνει εύκολα σε O(n) χρόνο.

Προφανώς, η πολυπλοκότητά σε χώρο είναι η ίδια, αφού για να αποθηκεύσουμε τις συναρτήσεις – σχέσεις, θα χρειαστεί να αποθηκεύσουμε όλες τις πλειάδες.

4.3 Απαλοιφή με κάδους και ισχυροί περιορισμοί

Εάν δοκιμάσουμε να λύσουμε το «ενισχυμένο» πρόβλημα για CAP με τον αλγόριθμο ELIM-OPT, θα υπάρξει πρόβλημα. Ο ELIM-OPT θα υπολογίσει λάθος συνάρτηση κόστους, αφού δε θα λάβει υπόψιν του, τους ισχυρούς περιορισμούς. Υπάρχουν δύο τρόποι να το αντιμετωπίσουμε.

4.3.1 Οι ισχυροί περιορισμοί μεταμφιεσμένοι σαν ασθενείς περιορισμοί

Ο πρώτος τρόπος είναι να θεωρήσουμε τους ισχυρούς περιορισμούς σαν ασθενείς. Για κάθε ισχυρό περιορισμό ορίζουμε μία συνάρτηση που για τις μη συνεπείς πλειάδες του επιστρέφει μηδέν και για τις συνεπείς επιστρέφει ένα πολύ μεγάλο θετικό ακέραιο. Αυτή η προσέγγιση θα μας επιστρέψει το σωστό αποτέλεσμα, αλλά μειονεκτεί στο ότι χάνονται τα χαρακτηριστικά των ισχυρών περιορισμών. Ιδιαίτερα, όταν παραλλάξουμε τον BE ώστε να γίνει πιο γρήγορος και να επιστρέφει κόστη κοντά στο βέλτιστο, θα χάσουμε την πληροφορία ποιοι είναι οι ισχυροί περιορισμοί καθώς αυτοί αποσαφηνίζουν τη δομή τους προβλήματος. Για αυτό το λόγο εξάλλου,

αποτιμούμε τους ισχυρούς περιορισμούς με ένα πολύ μεγάλο θετικό αριθμό[3].

Στο CAP είναι σημαντικό να διατηρηθούν οι ισχυροί περιορισμοί καθώς αν όχι, μια λύση που θα προσεγγίζει το βέλτιστο κόστος θα μπορεί να μας επιστρέψει δημοπρασίες που να μοιράζονται ένα προϊόν.

4.3.2 Παράδειγμα ELIM-OPT με μόνο ασθενείς περιορισμούς

Θα μετατρέψουμε τους ισχυρούς περιορισμούς σε ασθενείς για το «ενισχυμένο παράδειγμα CAP» και θα το λύσουμε! Πάντως, πρέπει να σημειώσουμε ότι δεν χρησιμοποιείται αυτή η προσέγγιση μεταμφίεσης μεταβλητών, αλλά την παραθέτουμε γιατί είναι μια καλή ευκαιρία να δούμε πως θα λειτουργούν οι BE, καθώς με αυτόν τον τρόπο θα έχουμε μόνο ασθενείς περιορισμούς.

Για ευκολία, παραθέτουμε εν συντομία το ενισχυμένο παράδειγμα CAP εδώ.

- ➔ $X = \{x_1,x_2,x_3,x_4,x_5,x_6\}$, όπου το x_i συμβολίζει τη δημοπρασία b_i
- ➔ $D = \{D_1,D_2,D_3,D_4,D_5,D_6\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = \{0,1\}$.
- ➔ Ασθενείς περιορισμοί C_s:
  - $F_1(x_1)=\begin{Bmatrix}8,\text{εάν } x_1=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_2(x_2)=\begin{Bmatrix}6,\text{εάν } x_2=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_3(x_3)=\begin{Bmatrix}5,\text{εάν } x_3=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_4(x_4)=\begin{Bmatrix}2,\text{εάν } x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_5(x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_6(x_6)=\begin{Bmatrix}4,\text{εάν } x_6=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{12}(x_1,x_2)=\begin{Bmatrix}10,\text{εάν } x_1=x_2=1\\8,\text{εάν } x_1=x_2=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{13}(x_1,x_3)=\begin{Bmatrix}8,\text{εάν } x_1=x_3=1\\2,\text{εάν } x_1=1 \text{ και } x_3=0\\10,\text{εάν } x_1=0 \text{ και } x_3=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix}8,\text{εάν } x_2=x_3=x_5\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix}8,\text{εάν } x_1=x_2=x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{56}(x_5,x_6)=\begin{Bmatrix}10,\text{εάν } x_5=x_6=1\\6,\text{εάν } x_5=x_6=0\\0,\text{αλλιώς}\end{Bmatrix}$
- ➔ Ισχυροί περιορισμοί, $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}=R_{13}=R_{14}=R_{24}=R_{25}=R_{35}=\{(1,0),(0,1),(0,0)\}$.

3 Στη βιβλιογραφία συνηθίζεται να αποτιμούμε τις έγκυρες πλειάδες με το άπειρο. Αλλά αυτή η τακτική έχει κάποια τεχνικά προβλήματα και δε θα ακολουθηθεί.

Θα μετατρέψουμε τους ισχυρούς περιορισμούς σε ασθενείς. Για μεγάλο θετικό ακέραιο θα πάρουμε το άθροισμα όλων των μέγιστων τιμών των συναρτήσεων – ασθενών περιορισμών. Θα μπορούσαμε να πάρουμε το άπειρο. Αλλά οι πράξεις με το άπειρο είναι περίεργες. Ας συμβολίσουμε αυτόν τον αριθμό με Θ. Έχουμε:

$\Theta = \sup(F_1(x_1)) + \sup(F_2(x_2)) + \sup(F_3(x_3)) + \sup(F_4(x_4)) + \sup(F_5(x_5)) + \sup(F_6(x_6)) + \sup(F_{12}(x_1,x_2)) + \sup(F_{13}(x_1,x_3)) + \sup(F_{235}(x_2,x_3,x_5)) + \sup(F_{124}(x_1,x_2,x_4)) + \sup(F_{56}(x_5,x_6)) = 8 + 6 + 5 + 2 + 2 + 4 + 10 + 10 + 8 + 8 + 10 = 73.$

Εύκολα μπορεί να δει κανείς ότι το κόστος M ποτέ δεν μπορεί να υπερβεί αυτή την ποσότητα Θ, καθώς,

$M = F_1(x_1) + F_2(x_2) + F_3(x_3) + F_4(x_4) + F_5(x_5) + F_6(x_6) + F_{12}(x_1,x_2) + F_{13}(x_1,x_3) + F_{235}(x_2,x_3,x_5) + F_{124}(x_1,x_2,x_4) + F_{56}(x_5,x_6) \leqslant$

$\leqslant \sup(F1(x1)) + \sup(F2(x2)) + \sup(F3(x3)) + \sup(F4(x4)) + \sup(F5(x5)) + \sup(F6(x6)) + \sup(F12(x1,x2)) + \sup(F13(x1,x3)) + \sup(F235(x2,x3,x5)) + \sup(F124(x1,x2,x4)) + \sup(F56(x5,x6)) = \Theta.$

Οπότε, ποτέ δε θα κινδυνεύσουμε να υπερβούν οι ασθενείς περιορισμοί τους ισχυρούς.

Μετατρέπουμε τους ισχυρούς περιορισμούς σε ασθενείς:

$$F_{12}(x_1,x_2)=\begin{Bmatrix}0, \text{εάν } x_1=x_2=1\\ 73, \text{αλλιώς}\end{Bmatrix}$$

$$F_{13}(x_1,x_3)=\begin{Bmatrix}0, \text{εάν } x_1=x_3=1\\ 73, \text{αλλιώς}\end{Bmatrix}$$

$$F_{14}(x_1,x_4)=\begin{Bmatrix}0, \text{εάν } x_1=x_4=1\\ 73, \text{αλλιώς}\end{Bmatrix}$$

$$F_{24}(x_2,x_4)=\begin{Bmatrix}0, \text{εάν } x_2=x_4=1\\ 73, \text{αλλιώς}\end{Bmatrix}$$

$$F_{25}(x_2,x_5)=\begin{Bmatrix}0, \text{εάν } x_2=x_5=1\\ 73, \text{αλλιώς}\end{Bmatrix}$$

$$F_{35}(x_3,x_5)=\begin{Bmatrix}0, \text{εάν } x_3=x_5=1\\ 73, \text{αλλιώς}\end{Bmatrix}$$

Ορισμένες από τις νέες συναρτήσεις έχουν την ίδια εμβέλεια με τους ασθενείς περιορισμούς. Συγχωνεύουμε και έχουμε:

$$F_{12}(x_1,x_2)=\begin{Bmatrix}10, \text{εάν } x_1=x_2=1\\ 81, \text{εάν } x_1=x_2=0\\ 73, \text{αλλιώς}\end{Bmatrix}$$

$$F_{13}(x_1,x_3)=\begin{Bmatrix}8, \text{εάν } x_1=x_3=1\\ 75, \text{εάν } x_1=1 \text{ και } x_3=0\\ 83, \text{εάν } x_1=0 \text{ και } x_3=0\\ 73, \text{αλλιώς}\end{Bmatrix}$$

Θα εκτελέσουμε τον ELIM-OPT κατά την απαρίθμηση $d = (x_1,x_3,x_2,x_5,x_4,x_6)$. Τοποθετούμε αρχικά τις συναρτήσεις σε κάδους.

Κάδος	Συναρτήσεις
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5,x_6)$
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $F_{14}(x_1,x_4)$, $F_{24}(x_2,x_4)$
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2,x_3,x_5)$, $F_{25}(x_2,x_5)$, $F_{35}(x_3,x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1,x_3)$
Bucket(x_1):	$F_1(x_1)$

Πίνακας 4: Οι κάδοι μετά την αρχικοποίηση του ELIM-OPT

Στο Bucket(x_6), θα υπολογίσουμε τη $h^{x_6}(x_5)=\begin{Bmatrix}14, \text{εάν } x_5=1 (x_6=1)\\ 6, \text{αλλιώς} (x_6=0)\end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_5).

Στο Bucket(x_4), θα υπολογίσουμε τη $h^{x_4}(x_1,x_2)=\begin{Bmatrix}148, \text{εάν } x_1=x_2=0 (x_4=1)\\ 146, \text{αλλιώς} (x_4=0)\end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_2).

Στο Bucket(x_5), θα υπολογίσουμε τη $h^{x_5}(x_2,x_3)=\begin{Bmatrix}162, \text{εάν } x_2=x_3=0 (x_5=1)\\ 152, \text{αλλιώς} (x_5=0)\end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_2).

Στο Bucket(x_2), θα υπολογίσουμε τη $h^{x_2}(x_1,x_3)=\begin{Bmatrix}369, \text{εάν } x_1=x_3=1 (x_2=0)\\ 381, \text{εάν } x_1=1 \text{ και } x_3=0 (x_2=0)\\ 391, \text{εάν } x_1=0 \text{ και } x_3=0 (x_2=0)\\ 379, \text{αλλιώς} (x_2=0)\end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_3).

Στο Bucket(x_3), θα υπολογίσουμε τη $h^{x_3}(x_1)=\begin{Bmatrix}474, \text{εάν } x_1=0 (x_3=0)\\ 458, \text{αλλιώς} (x_3=0)\end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_1).

Στο Bucket(x_1), θα υπολογίσουμε τη ψεύτικη συνάρτηση κόστος, M = 474(x_1 = 0).

Οι κάδοι:

Κάδος	Συναρτήσεις
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5,x_6)$
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $F_{14}(x_1,x_4)$, $F_{24}(x_2,x_4)$
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2,x_3,x_5)$, $F_{25}(x_2,x_5)$, $F_{35}(x_3,x_5)$, $h^{x_6}(x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $h^{x_4}(x_1,x_2)$, $h^{x_5}(x_2,x_3)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1,x_3)$, $h^{x_2}(x_1,x_3)$,
Bucket(x_1):	$F_1(x_1)$, $h^{x_3}(x_1)$

Πίνακας 5: Οι κάδοι μετά το πέρας του ELIM-OPT

Σε αυτή τη M έχουν συμπεριληφθεί οι μεταμφιεσμένοι περιορισμοί. Δε μας πειράζει, συνεχίζουμε τον ELIM-OPT.

Στο Bucket(x_1), πρέπει $x_1 = 0$.
Στο Bucket(x_3), πρέπει $x_3 = 0$.
Στο Bucket(x_2), πρέπει $x_2 = 0$.
Στο Bucket(x_5), πρέπει $x_5 = 1$.
Στο Bucket(x_4), πρέπει $x_4 = 1$.
Στο Bucket(x_6), πρέπει $x_6 = 1$.

Ας υπολογίσουμε και την πραγματική συνάρτηση κόστους. Κοιτώντας τις αρχικές συναρτήσεις F παίρνουμε για $M = F_1(0) + F_2(0) + F_3(0) + F_4(1) + F_5(1) + F_6(1) + F_{12}(0,0) + F_{13}(0,0) + F_{235}(0,0,1) + F_{124}(0,0,1) + F_{56}(1,1) = 0 + 0 + 0 + 2 + 2 + 4 + 8 + 10 + 0 + 0 + 10 = 36$

4.3.3 Αλγόριθμος ELIM-OPT-CONS

Η παραπάνω μεταμφίεση των ισχυρών περιορισμών σε ασθενείς δεν είναι καθόλου εύχρηστη. Η διάκριση μεταξύ ασθενών και ισχυρών περιορισμών δεν υπάρχει. Όταν θα θέλουμε να διαφοροποιήσουμε λίγο τους BE, δε θα μπορούμε, καθώς μη λαμβάνοντας υπόψιν έστω και ένα ισχυρό περιορισμό, αλλοιώνεται η φύση του προβλήματος. Μπορούμε να συγχωρήσουμε λιγότερο κέρδος, αλλά δεν μπορούμε να πουλήσουμε το ίδιο προϊόν σε παραπάνω από μία δημοπρασίες.

Η λύση στο παραπάνω πρόβλημα είναι εύκολη. Ο αλγόριθμος ELIM-OPT-CONS τρέχει τον ELIM-OPT ενώ παράλληλα ελέγχει για ισχυρούς περιορισμούς.

Αλγόριθμος ELIM-OPT-CONS

Δεδομένα:

Ένα δίκτυο με περιορισμούς (X,D,C_h,C_s), με $C_h=\{R_{S_1},\dots,R_{S_l}\}, C_s=\{F_{Q_1},\dots,F_{Q_m}\}$, l = |C_h| και m = |C_s|.

Μία απαρίθμηση d των μεταβλητών X.

Αρχή Αλγορίθμου:

Για κάθε ισχυρό περιορισμό R_{S_i} , με i από το 1 στο l, **κάνε:**

S_i = η εμβέλεια του R_{S_i}

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη S_i

Τοποθέτησε τον R_{S_i} στον κάδο Bucket(x_b)

Τέλος Για κάθε

Για κάθε ασθενή περιορισμό F_{Q_i} , με i από το 1 στο m, **κάνε:**

Q_i = η εμβέλεια της F_{Q_i}

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη Q_i

Τοποθέτησε τη F_{Q_i} στον κάδο Bucket(x_b)

Τέλος Για κάθε

Για κάθε κάδο Bucket(x_p), με p από το n στο 1, **κάνε:**

$R_{S_1},\dots,R_{S_k}$ = οι ισχυροί περιορισμοί που υπάρχουν στο Bucket(x_p)

$S_1,\dots,S_k$ = οι εμβέλειες των $R_{S_1},\dots,R_{S_k}$

$U_p = (\ \bigcup_i S_i\ ) - \{x_p\}$

$R^{x_p}=\pi_{U_p}$ JOINT $\underset{i=1}{\overset{k}{}} R_i$,σχημάτισε τον καθολικό περιορισμό του Bucket(x_p)

$h^{x_1},\dots,h^{x_j}$ = οι συναρτήσεις-περιορισμοί που έχουν προστεθεί στο Bucket(x_p)

$Q_1,\dots,Q_j$ = οι εμβέλειες των $h^{x_1},\dots,h^{x_j}$

$F_{Q_1},\dots,F_{Q_o}$ = οι αρχικοί ασθενείς περιορισμοί που υπάρχουν στο Bucket(x_p)

$Q_1,\dots,Q_o$ = οι εμβέλειες των $F_{Q_1},\dots,F_{Q_o}$

$V_p = (\ \bigcup_i Q_i\ ) - \{x_p\}$

$W_p = U_p \ \cup\ V_p$

Για κάθε πλειάδα t του W_p **κάνε:**

$$h^{x_p}(t)=\max_{a_p\text{έ.ώ.}R^{x_p}(t,a_p)}\{\sum_{i=1}^{j}h^{x_i}(t,a_p)+\sum_{i=1}F_{Q_i}(t,a_p)\}$$

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στο W_p

Τοποθέτησε τη h^{x_p} στο Bucket(x_b)

x_g = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στο U_p

Τοποθέτησε τη R^{x_p} στο Bucket(x_g)

Τέλος Για κάθε

Τέλος Για κάθε

Για i από το 1 στο n, **κάνε**

Δεδομένης αποτίμησης $\vec{a_{i-1}}$, αποτίμησε το x_i με μια τιμή a_i ε D_i που μεγιστοποιεί το άθροισμα των συναρτήσεων στο Bucket(x_i)

Τέλος Για

Τέλος Αλγορίθμου

Έξοδος:

Μια συνεπής αποτίμηση που αποφέρει το μέγιστο κόστος $\sum_{F_i\in C_S} F_i$.

4.3.4 Πολυπλοκότητα ELIM-OPT-CONS

Από ότι θα έχετε καταλάβει και εσείς, ο ELIM-OPT-CONS είναι παρόμοιος με τον ELIM-OPT και έχουν την ίδια πολυπλοκότητα.

Θεώρημα: Πολυπλοκότητα ELIM-OPT-CONS

Η πολυπλοκότητά σε χώρο και χρόνο του ELIM-OPT-CONS για μια απαρίθμηση d, είναι $O(r \cdot k^{w^*(d)})$ *, όπου k είναι η πληθικότητα της μεταβλητής με το μεγαλύτερο σε πληθικότητα πεδίο τιμών και r είναι το πλήθος των ασθενών περιορισμών.*

Απόδειξη:

Η απόδειξη δεν παρατίθεται, καθώς είναι η ίδια με του ELIM-OPT.

4.3.5 Παράδειγμα ELIM-OPT-CONS

Επιτέλους, θα τρέξουμε τον ELIM-OPT-CONS σε αριθμητικά δεδομένα! Όταν εκτελέσαμε τον ELIM-OPT για τους μεταμφιεσμένους περιορισμούς, παραβλέψαμε τη φύση των ισχυρών περιορισμών, περιορίζοντας είτε την τροποποίηση του αλγορίθμου ώστε να πάρουμε προσεγγιστικά καλά αποτελέσματα, είτε την ενσωμάτωση του σε άλλες τεχνικές στο «χώρο».

Για ευκολία, παραθέτουμε εν συντομία το ενισχυμένο παράδειγμα CAP εδώ.

- ➔ $X = \{x_1,x_2,x_3,x_4,x_5,x_6\}$, όπου το x_i συμβολίζει τη δημοπρασία b_i
- ➔ $D = \{D_1,D_2,D_3,D_4,D_5,D_6\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = \{0,1\}$.
- ➔ Ασθενείς περιορισμοί C_s:
  - $F_1(x_1)=\begin{Bmatrix}8,\text{εάν } x_1=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_2(x_2)=\begin{Bmatrix}6,\text{εάν } x_2=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_3(x_3)=\begin{Bmatrix}5,\text{εάν } x_3=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_4(x_4)=\begin{Bmatrix}2,\text{εάν } x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_5(x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_6(x_6)=\begin{Bmatrix}4,\text{εάν } x_6=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{12}(x_1,x_2)=\begin{Bmatrix}10,\text{εάν } x_1=x_2=1\\8,\text{εάν } x_1=x_2=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{13}(x_1,x_3)=\begin{Bmatrix}8,\text{εάν } x_1=x_3=1\\2,\text{εάν } x_1=1 \text{ και } x_3=0\\10,\text{εάν } x_1=0 \text{ και } x_3=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix}8,\text{εάν } x_2=x_3=x_5\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix}8,\text{εάν } x_1=x_2=x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$

- $F_{56}(x_5, x_6) = \begin{Bmatrix} 10, \text{εάν } x_5 = x_6 = 1 \\ 6, \text{εάν } x_5 = x_6 = 0 \\ 0, \text{αλλιώς} \end{Bmatrix}$

➔ Ισχυροί περιορισμοί, $C_h = \{R_{12}, R_{13}, R_{14}, R_{24}, R_{25}, R_{35}\}$, όπου $R_{12}=R_{13}=R_{14}=R_{24}=R_{25}=R_3 = \{(1,0), (0,1), (0,0)\}$.

Θα εκτελέσουμε τον ELIM-OPT-CONS κατά την απαρίθμηση d = $(x_1, x_3, x_2, x_5, x_4, x_6)$. Τοποθετούμε αρχικά τους ασθενείς και ισχυρούς περιορισμούς σε κάδους.

Κάδος	Συναρτήσεις
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5, x_6)$
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1, x_2, x_4)$, $R_{14}(x_1, x_4)$, $R_{24}(x_2, x_4)$
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2, x_3, x_5)$, $R_{25}(x_2, x_5)$, $R_{35}(x_3, x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1, x_2)$, $R_{12}(x_1, x_2)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1, x_3)$, $R_{13}(x_1, x_3)$
Bucket(x_1):	$F_1(x_1)$

Πίνακας 6: Οι κάδοι μετά την αρχικοποίηση του ELIM-OPT-CONS

Στο Bucket(x_6), θα υπολογίσουμε τη $h^{x_6}(x_5) = \begin{Bmatrix} 14, \text{εάν } x_5 = 1 (x_6 = 1) \\ 6, \text{αλλιώς} (x_6 = 0) \end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_5).

Στο Bucket(x_4), πρώτα, θα υπολογίσουμε τη $R^{x_4}(x_1, x_2) = \{(0,0), (0,1), (1,0), (1,1)\}$. Επειδή η $R^{x_4}(x_1, x_2)$ είναι καθολικός περιορισμός, δε θα την τοποθετήσουμε στο Bucket(x_2), δεν ασχολούμαστε με καθολικούς περιορισμούς.

Έπειτα θα υπολογίσουμε τη $h^{x_4}(x_1, x_2) = \begin{Bmatrix} 2, \text{εάν } x_1 = x_2 = 0 (x_4 = 1) \\ 0, \text{αλλιώς} (x_4 = 0) \end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_2).

Στο Bucket(x_5) πρώτα θα υπολογίσουμε τη $R^{x_5}(x_2, x_3) = \{(0,0), (0,1), (1,0), (1,1)\}$, αλλά δε θα την τοποθετήσουμε στο Bucket(x_2).

Ύστερα θα υπολογίσουμε τη $h^{x_5}(x_2, x_3) = \begin{Bmatrix} 16, \text{εάν } x_2 = x_3 = 0 (x_5 = 1) \\ 6, \text{αλλιώς} (x_5 = 0) \end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_2).

Στο Bucket(x_2), πρώτα, θα υπολογίσουμε τη $R^{x_2}(x_1) = \{0,1\}$, αλλά δε θα την τοποθετήσουμε στο Bucket(x_1).

Κατόπιν θα υπολογίσουμε τη $h^{x_2}(x_1, x_3) = \begin{Bmatrix} 6, \text{εάν } x_1 = x_3 = 1 (x_2 = 0) \\ 16, \text{εάν } x_1 = 1 \text{ και } x_3 = 0 (x_2 = 0) \\ 26, \text{εάν } x_1 = 0 \text{ και } x_3 = 0 (x_2 = 0) \\ 16, \text{αλλιώς} (x_2 = 0) \end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_3).

Στο Bucket(x_3), πρώτα, θα υπολογίσουμε τη $R^{x_3}(x_1) = \{0,1\}$, αλλά δε θα την τοποθετήσουμε στο Bucket(x_1).

Στο Bucket(x_3), θα υπολογίσουμε τη $h^{x_3}(x_1) = \begin{Bmatrix} 36, \text{εάν } x_1 = 0 (x_3 = 0) \\ 18, \text{αλλιώς} (x_3 = 0) \end{Bmatrix}$ και θα την τοποθετήσουμε στο Bucket(x_1).

Στο Bucket(x_1), θα υπολογίσουμε τη $h^{x_1} = 36, (x_1 = 0)$. Οπότε το βέλτιστο κόστος είναι 36. Θυμηθείτε ότι 36 βρήκαμε και με τη μέθοδο της μεταμφίεσης των ισχυρών περιορισμών.

Οι κάδοι:

Κάδος	Συναρτήσεις
Bucket(x_6):	$F_6(x_6)$, $F_{56}(x_5,x_6)$
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $R_{14}(x_1,x_4)$, $R_{24}(x_2,x_4)$
Bucket(x_5):	$F_5(x_5)$, $F_{235}(x_2,x_3,x_5)$, $R_{25}(x_2,x_5)$, $R_{35}(x_3,x_5)$, $h^{x_6}(x_5)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $R_{12}(x_1,x_2)$, $h^{x_4}(x_1,x_2)$, $h^{x_5}(x_2,x_3)$
Bucket(x_3):	$F_3(x_3)$, $F_{13}(x_1,x_3)$, $R_{13}(x_1,x_3)$, $h^{x_2}(x_1,x_3)$
Bucket(x_1):	$F_1(x_1)$ h^{x_1}

Πίνακας 7: Οι κάδοι μετά το τέλος του ELIM-OPT-CONS

Συνεχίζουμε τον ELIM-OPT-CONS, για να πάρουμε την αποτίμηση που μας απέφερε το βέλτιστο κόστος.
Στο Bucket(x_1), πρέπει $x_1 = 0$.
Στο Bucket(x_3), πρέπει $x_3 = 0$.
Στο Bucket(x_2), πρέπει $x_2 = 0$.
Στο Bucket(x_5), πρέπει $x_5 = 1$.
Στο Bucket(x_4), πρέπει $x_4 = 1$.
Στο Bucket(x_6), πρέπει $x_6 = 1$.

Πρέπει να είμαστε εκστασιασμένοι αυτήν τη στιγμή. Λύσαμε ένα COP με έναν αλγόριθμο που είναι πλήρης και έγκυρος. Παρόλα αυτά, δεν έχουμε καταφέρει κάτι το τρομερό. Ο ELIM-OPT-CONS είναι αργός και απαιτεί πολύ χώρο, $O(r \cdot k^{w^*(d)})$. Με μεγάλα δεδομένα, όπως είναι αυτά των πραγματικών προβλημάτων που υπάρχουν, ο ELIM-OPT-CONS είναι καταδικασμένος να εκτελείται για πάντα. Τυχερά είναι μόνο τα προβλήματα, που έχουν μικρό επαγόμενο πλάτος για την απαρίθμηση που επιλέχθηκε.

Δε θα είχαμε κάνει τόσο κόπο, εάν δεν υπήρχε κάτι που να μας τραβήξει από το αδιέξοδο της πολυπλοκότητας. Διάφορες παραλλαγές του ELIM-OPT-CONS μπορούν να είναι γρήγορες, ελαφρές στην κατανάλωση χώρου και να βρίσκουν κόστη πολύ κοντά στο βέλτιστο. Τέτοιες μεθόδους θα δούμε παρακάτω.

4.4 BE ∈ δυναμικός προγραμματισμός

Έχουμε παρατηρήσει το εξής: Το πρόβλημά μας ανάγεται στο να υπολογίσουμε έξι ίδια προβλήματα που έχουν μικρότερη είσοδο, τους έξι κάδους που δημιουργούνται. Είναι σαν έχουμε έξι μικρότερα COP. Αυτή η τεχνική λέγεται δυναμικός προγραμματισμός (dynamic programming). Στην πραγματικότητα, είναι και άλλα χαρακτηριστικά που μας δείχνουν ότι έχουμε δυναμικό προγραμματισμό, (τα υποπροβλήματα είναι επικαλυπτόμενα (overlapping), κτλ), αλλά μας ενδιαφέρει το συγκεκριμένο που τονίζουμε. Σε αυτού του είδους τη μεθοδολογία, την πολυπλοκότητα την αποφασίζει κυρίως το μεγαλύτερο σε είσοδο από τα υποπροβλήματα. Για να βελτιώσουμε τους BE πρέπει να οδηγήσουμε τη VE ούτως ώστε το μεγαλύτερο υποπρόβλημα να έχει τη μικρότερη είσοδο. Πώς θα το επιτύχουμε αυτό;

4.5 VE με διαφορετική απαρίθμηση

Η απάντηση στο ερώτημα του προηγούμενου εδαφίου, η ελαχιστοποίηση της εισόδου στο υποπρόβλημα με τη μεγαλύτερη είσοδο επιτυγχάνεται με την ελαχιστοποίηση των ορισμάτων των νέων συναρτήσεων – ασθενών περιορισμών που δημιουργούνται. Για να το επιτύχουμε αυτό, πρέπει να διαλέξουμε μία άλλη απαρίθμηση με την οποία θα επιλέγουμε τις μεταβλητές μας.

Για το συγκεκριμένο πρόβλημα δεν υπάρχει καλύτερη απαρίθμηση, αλλά για να γίνει κατανοητό το τι εννοούμε, θα επιδείξουμε μία απαρίθμηση που θα μας οδηγήσει σε ασθενείς περιορισμούς με περισσότερα ορίσματα. Ας δούμε μία τέτοια απαρίθμηση. Διαλέγουμε τη d_2 =

$(x_1,x_5,x_4,x_3,x_2,x_6)$. Ξεκινάμε τις πράξεις, περιθωριοποιώντας πρώτα τη x_6:

$$M=\max_{x_1,x_2,x_3,x_4,x_5,x_6} F=\max_{x_1,x_2,x_3,x_4,x_5,x_6}\sum F_i=$$

$$=\max_{x_1,x_2,x_3,x_4,x_5,x_6}\{F_1(x_1)+F_2(x_2)+F_3(x_3)+F_4(x_4)+F_5(x_5)+F_6(x_6)+$$

$$+F_{12}(x_1,x_2)+F_{13}(x_1,x_3)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+F_{56}(x_5,x_6)\}=$$

$$=\max_{x_1,x_2,x_3,x_4,x_5}\{F_1(x_1)+F_2(x_2)+F_3(x_3)+F_4(x_4)+F_5(x_5)+F_{12}(x_1,x_2)+$$

$$+F_{13}(x_1,x_3)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}$$

Δεν αλλάζει κάτι σε σχέση με τα προηγούμενα, τώρα όμως επιλέγουμε τη x_2.

$$M=\max_{x_1,x_3,x_4,x_5}\{F_1(x_1)+F_3(x_3)+F_4(x_4)+F_5(x_5)+F_{13}(x_1,x_3)+$$

$$+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}=$$

Επιλέγουμε τη x_3.

$$M=\max_{x_1,x_4,x_5}\{F_1(x_1)+F_4(x_4)+F_5(x_5)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+$$

$$+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}\}$$

Τελικά, κάνοντας τις ίδιες πράξεις και για τις υπόλοιπες μεταβλητές έχουμε:

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_5}\{F_5(x_5)+\max_{x_4}\{F_4(x_4)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+$$

$$+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}\}\}\}\}\}$$

Ορίζουμε τις νέες συναρτήσεις και έχουμε:

$$h^{x_6}(x_5)=\max_{x_6}\{F_6(x_6)+F_{56}(x_5,x_6)\}$$

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_5}\{F_5(x_5)+\max_{x_4}\{F_4(x_4)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+$$

$$+\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)+h^{x_6}(x_5)\}\}\}\}\}$$

$$h^{x_2}(x_1,x_3,x_4,x_5)=\max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)+F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)\}$$

.

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_5}\{F_5(x_5)+\max_{x_4}\{F_4(x_4)+\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+$$

$$+h^{x_2}(x_1,x_3,x_4,x_5)+h^{x_6}(x_5)\}\}\}\}$$

$$h^{x_3}(x_1,x_4,x_5)=\max_{x_3}\{F_3(x_3)+F_{13}(x_1,x_3)+h^{x_2}(x_1,x_3,x_4,x_5)\}$$

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_5}\{F_5(x_5)+\max_{x_4}\{F_4(x_4)+h^{x_3}(x_1,x_4,x_5)+h^{x_6}(x_5)\}\}\}$$

$$h^{x_4}(x_1,x_5)=\max_{x_4}\{F_4(x_4)+h^{x_3}(x_1,x_4,x_5)\}$$

$$M=\max_{x_1}\{F_1(x_1)+\max_{x_5}\{F_5(x_5)+h^{x_4}(x_1,x_5)+h^{x_6}(x_5)\}\}$$

$$h^{x_5}(x_1)=\max_{x_5}\{F_5(x_5)+h^{x_4}(x_1,x_5)+h^{x_6}(x_5)\}$$

$$M=\max_{x_1}\{F_1(x_1)+h^{x_5}(x_1)\}$$

$$h^{x_1}=\max_{x_1}\{F_1(x_1)+h^{x_5}(x_1)\}$$

$$M=h^{x_1}$$

Το ότι επιλέξαμε διαφορετική απαρίθμηση δεν μας οδήγησε σε λάθος αποτέλεσμα, βρίσκουμε ακριβώς το ίδιο κόστος. Ωστόσο, οι συναρτήσεις που δημιουργούνται έως ότου να φτάσουμε στο τελικό αποτέλεσμα δεν είναι οι ίδιες. Παρατηρούμε ότι η $h^{x_2}(x_1,x_3,x_4,x_5)$ είναι ασθενής περιορισμός ορισμένος πάνω σε τέσσερις μεταβλητές. Αυτό οδηγεί σε συνένωση τεσσάρων μεταβλητών, ενώ με τη απαρίθμηση d_1 είχαμε συναρτήσεις με το πολύ δύο ορίσματα. Αυτή η μείωση των ορισμάτων στις συναρτήσεις μειώνει την πολυπλοκότητα του υπολογισμού τους και επομένως, και των ΒΕ. Θα θεμελιώσουμε αυτή τη συμπεριφορά των ΒΕ ανάλογα με τις απαριθμήσεις στο επόμενο ακριβώς εδάφιο.

4.5.1 Απαραίτητη γραφοθεωρία

4.5.1.1 Γράφοι με περιορισμούς

Για να μελετήσουμε την επίδραση της επιλογής μιας απαρίθμησης στους ΒΕ, πρέπει πρώτα να παραθέσουμε κάποιους ορισμούς και κάποια θεωρήματα. Αφού το κάνουμε, θα εξηγήσουμε το τι επιπτώσεις είναι δυνατόν να έχει η επιλογή μιας απαρίθμησης στους ΒΕ.

Αρχικά θα αναφερθούμε στα CP και μετά θα επεκταθούμε στα COP. Ξεκινάμε με τρεις ορισμούς.

Ορισμός: N-αδικός περιορισμός

> *N-αδικό (n-ary) καλούμε τον περιορισμό που στην εμβέλειά του έχει ακριβώς n διαφορετικές μεταβλητές.*

Σημειώνουμε ότι ανάλογα ορίζουμε και ν-αδικές σχέσεις.

Ορισμός: Καθολικός περιορισμός

> *Καθολικό (universal) καλούμε τον περιορισμό που ισούται με το καρτεσιανό γινόμενο των πεδίων τιμών των μεταβλητών που συμμετέχουν στον περιορισμό,* $C_{x_i,\dots,x_i}\subseteq D_i x \dots x D_j$.

Σημειώνουμε ότι ανάλογα ορίζουμε και καθολικές σχέσεις.

Ορισμός: Γράφος με περιορισμούς

> *Γράφος με περιορισμούς (constraint graph) για ένα CP με δυαδικούς περιορισμούς είναι ένας μη κατευθυνόμενος γράφος που έχει σαν κόμβους μεταβλητές του CP. Μεταξύ δύο κόμβων υπάρχει ακμή, εφόσον αυτές οι μεταβλητές είναι στην εμβέλεια (scope) ενός δυαδικού περιορισμού. Πρέπει να τονίσουμε ότι περιορισμούς που επιτρέπουν όλες τις δυνατές αποτιμήσεις σε όλες τις μεταβλητές της εμβέλειάς τους, δεν τους αναπαριστούμε με ακμές στο γράφο.*

Ο γράφος με περιορισμούς, μας βοηθά στο να μελετήσουμε καλύτερα το δικτύου με περιορισμούς, που αντιπροσωπεύει. Στον ορισμό, αναφερθήκαμε σε δίκτυα με δυαδικούς περιορισμούς μόνο και όλα τα αποτελέσματα στο χώρο αφορούν τέτοια δίκτυα. Αν και δεν είναι λίγα τα δίκτυα αυτά, τα περισσότερα που συναντούμε σε πραγματικά προβλήματα έχουν περιορισμούς με περισσότερα από δύο ορίσματα. Μπορούμε να ορίσουμε και για αυτά τα δίκτυα γράφους με περιορισμούς και το κάνουμε ως εξής.

Ορισμός: Υπεργράφος

> *Υπεργράφος (hypergraph) είναι μία δομή (structure) , H = (X, S). Με X συμβολίζουμε το σύνολο των μεταβλητών. S είναι ένα σύνολο υποσυνόλων από μεταβλητές,* $S=\{S_1,\ldots,S_l\}, S_i \subseteq X$ *και το καλούμε υπερακμές (hyperedges). Σε κάθε υποσύνολο μεταβλητών αντιστοιχούμε μία υπερακμή. Η υπερακμή είναι το αντίστοιχο των ακμών στους γράφους με περιορισμούς, μόνο που αντί να συνδέονται μέσω μιας ακμής δύο μεταβλητές – κόμβοι, συνδέονται πολλές μεταβλητές με μία μόνο ακμή!*

Οπότε, εάν θέλουμε να κατασκευάσουμε ένα γράφο με περιορισμούς για ένα δίκτυο με περιορισμούς ορισμένους σε περισσότερες από δύο μεταβλητές, απλά κατασκευάζουμε έναν υπεργράφο.

Μπορούμε να μετατρέψουμε τον υπεργράφο με περιορισμούς σε γράφο με περιορισμούς. Αυτό γίνεται με την εξής διαδικασία: για κάθε υπερακμή του υπεργράφου δημιουργούμε έναν κόμβο στο νέο γράφο. Συνδέουμε με ακμή κόμβους στο νέο γράφο που αντιστοιχούν σε υπερακμές που περιέχουν έστω και μία κοινή μεταβλητή.

Στη βιβλιογραφία τους δύο γράφους θα τους βρούμε σαν πρωτεύον και δυϊκό. Πρωτεύον τον αρχικό γράφο και δυϊκό τον καινούργιο γράφο.

Με αυτό τον τρόπο, εάν ένα δίκτυο δεν έχει μόνο δυαδικούς περιορισμούς, μπορούμε να κατασκευάσουμε τον υπεργράφο με περιορισμούς και μετά το δυϊκό του γράφο και να συνεχίσουμε να δουλεύουμε στο δυϊκό γράφο.

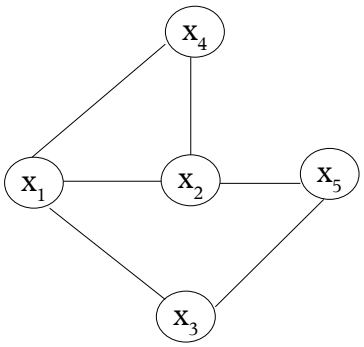
4.5.1.2 Γράφος με περιορισμούς για στιγμιότυπο του AP.

Θα χρησιμοποιήσουμε το στιγμιότυπο A του AP που έχουμε παραθέσει νωρίτερα. Για ευκολία το περιγράφουμε εν συντομία και εδώ:

Έχουμε:

- $C = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12} = R_{13} = R_{14} = R_{24} = R_{25} = R_{35} = \{(1,0),(0,1),(0,0)\}$.
- $X = \{x_1,x_2,x_3,x_4,x_5\}$, όπου το x_i συμβολίζει τη δημοπρασία b_i
- $D = \{D_1,D_2,D_3,D_4,D_5\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = \{0,1\}$.

Ο γράφος με περιορισμούς έχει πέντε κόμβους, ένα για κάθε μία μεταβλητή. Υπάρχουν ακμές μεταξύ των μεταβλητών που συμμετέχουν στον ίδιο περιορισμό. Υπάρχει ακμή μεταξύ της μεταβλητής x_1 και της μεταβλητής x_2, κτλ. Στο επόμενο σχήμα παρατίθεται ο γράφος περιορισμού για το στιγμιότυπο.



Σχήμα 3: Γράφος με περιορισμούς για το στιγμιότυπο Α του AP

4.5.1.3 Παρατήρηση

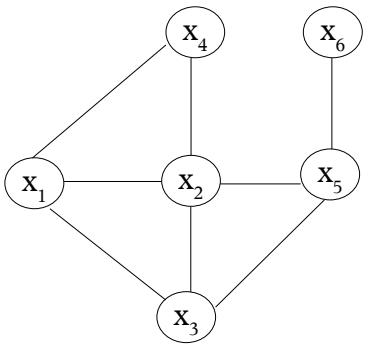
Οι γράφοι με περιορισμούς για COP είναι οι ίδιοι με αυτών των CP, αφού οι ασθενείς περιορισμοί ενός COP είναι περιορισμοί, και απεικονίζονται στο γράφο με περιορισμούς με τον ίδιο τρόπο με τους περιορισμούς ενός CP.

Στη βιβλιογραφία, οι γράφοι με περιορισμούς αναφέρονται και σαν γράφοι με κόστη (cost graphs). Ωστόσο, θα κρατήσουμε την ορολογία «γράφος με περιορισμούς» καθώς ο όρος «γράφος με κόστη» αναφέρεται σε κόστη που συνήθως παραπέμπουν σε ασθενείς περιορισμούς και όχι σε ισχυρούς περιορισμούς. Παρόλα αυτά οφείλουμε να σημειώσουμε ότι οι ισχυροί περιορισμοί μπορούν να θεωρηθούν σαν κόστη και να μετατραπούν σε ασθενείς περιορισμούς οπότε και να δικαιολογηθεί ο όρος «γράφος με κόστη».

Πριν συνεχίσουμε, πρέπει να αναφέρουμε κάτι πολύ σημαντικό! Στο ενισχυμένο CAP παράδειγμά μας, υπάρχουν ασθενείς περιορισμοί ορισμένοι σε τρεις μεταβλητές. Για να είμαστε απόλυτα σωστοί, θα έπρεπε να κατασκευάσουμε έναν υπεργράφο, να υπολογίσουμε το δυϊκό του και μετά να συνεχίσουμε με την ανάλυσή μας. Δε θα το κάνουμε όμως αυτό για τον εξής λόγο. Έχουμε ήδη εκτελέσει BE για το πρόβλημα με την πρωτεύουσα μοντελοποίηση. Εάν κατασκευάσουμε τον υπεργράφο και τον δυϊκό του γράφο, τότε θα πρέπει να εκτελέσουμε BE στη νέα δυϊκή μοντελοποίηση και δε θα μπορέσουμε να δούμε καθαρά τα αποτελέσματα που προκύπτουν για την επιλογή της απαρίθμησής. Καταχρηστικά, για κάθε περιορισμό με εμβέλεια τρεις μεταβλητές, θα κατασκευάζουμε μία κλίκα για τις τρεις μεταβλητές αυτές. Αυτό δεν είναι τελείως λάθος, καθώς σε μερικές περιπτώσεις αυτή η αντιμετώπιση είναι σωστή και ισοδύναμη με αυτή του υπεργράφου, αλλά δυστυχώς μόνο σε μερικές και όχι σε όλες.

Θα σχεδιάσουμε το γράφο με περιορισμούς για το ενισχυμένο CAP παράδειγμά μας. Ο γράφος με περιορισμούς έχει έξι κόμβους, ένα για κάθε μεταβλητή. Για κάθε ισχυρό και ασθενή περιορισμό έχει ακμές μεταξύ των μεταβλητών που συμμετέχουν στον περιορισμό αυτό. Για παράδειγμα, θα υπάρχει ακμή μεταξύ του κόμβου x_1 και του κόμβου x_2 λόγω του R_{12} και μεταξύ του x_5 και του x_6 λόγω του $F_{56}(x_5,x_6)$.

Ζωγραφίζουμε το γράφο με περιορισμούς:



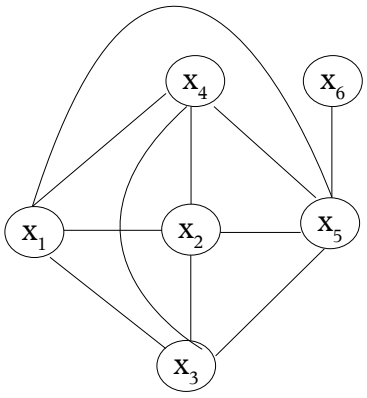
Σχήμα 4: Γράφος με περιορισμούς για το ενισχυμένο στιγμιότυπο του CAP κατά την απαρίθμηση d_1

4.5.1.4 ΒΕ σε γράφο με περιορισμούς

Πριν δοκιμάσουμε να τρέξουμε ένα ΒΕ στο παραπάνω γράφο με περιορισμούς για το ενισχυμένο στιγμιότυπο του CAP, καλό θα ήταν να ξαναδείτε την προηγούμενη παρατήρηση.

Θα τον τρέξουμε με την απαρίθμηση $d_1 = (x_1,x_3,x_2,x_5,x_4,x_6)$. Συμβουλευόμενοι το «Σχήμα 2: ΒΕ για το ενισχυμένο στιγμιότυπο CAP», αρχικά δημιουργείται η $h^{x_6}(x_5)$. Αυτός ο περιορισμός δε προσθέτει καμία ακμή στο γράφο, καθώς αναφέρεται σε μία μόνο μεταβλητή. Μετά δημιουργείται η $h^{x_4}(x_1,x_2)$. Αυτή επιβάλει μία ακμή μεταξύ των κόμβων x_1 και x_2, αλλά αυτή υπάρχει ήδη. Ύστερα παίρνουμε τη $h^{x_5}(x_2,x_3)$ και πάλι υπάρχει ήδη η ακμή $<x_2,x_3>$. Έπειτα, έχουμε τη $h^{x_2}(x_1,x_3)$ που δεν προσθέτει πάλι τίποτα καινούριο. Προφανώς, καμία αλλαγή στο γράφο δεν επιφέρουν και οι $h^{x_3}(x_1)$ και h^{x_1} .

Ας εκτελέσουμε ένα ΒΕ για τη δεύτερη απαρίθμηση που χρησιμοποιήσαμε, τη $d_2 = (x_1,x_5,x_4,x_3,x_2,x_6)$. Αρχικά θα πάρουμε τη $h^{x_6}(x_5)$, η οποία δεν επηρεάζει το γράφο μας. Μετά έχουμε τη $h^{x_2}(x_1,x_3,x_4,x_5)$, με της οποίας την εμφάνιση δημιουργούνται οι ακμές $<x_1,x_3>$, $<x_1,x_4>$, $<x_1,x_5>$, $<x_3,x_4>$, $<x_3,x_5>$ και $<x_4,x_5>$. Παρατηρούμε ότι οι $<x_1,x_5>$, $<x_3,x_4>$ και $<x_4,x_5>$ δεν προϋπήρχαν στο γράφο και προσαρτώνται σε αυτόν. Μετά προσθέτουμε τη $h^{x_3}(x_1,x_4,x_5)$. Από αυτήν δημιουργούνται οι $<x_1,x_5>$ και $<x_4,x_5>$ που υπάρχουν ήδη στο γράφο. Προφανώς, καμία αλλαγή δεν επιφέρουν και οι $h^{x_4}(x_1,x_5)$, $h^{x_5}(x_1)$ και h^{x_1} . Ο γράφος που δημιουργείται φαίνεται στο επόμενο σχήμα:

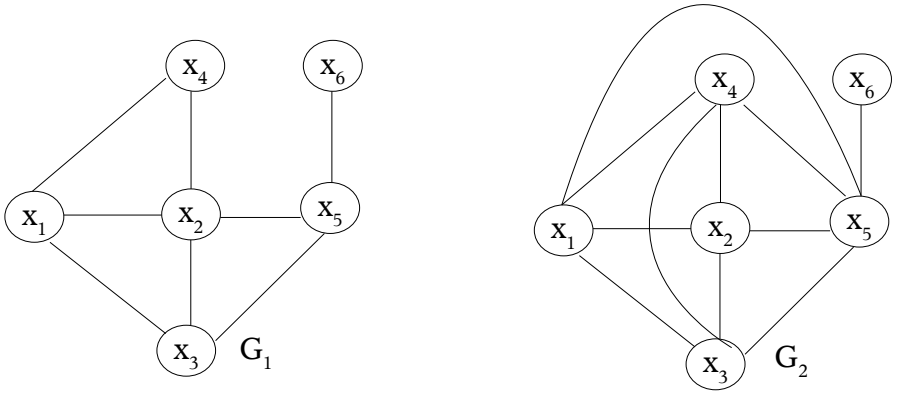


Σχήμα 5: Γράφος με περιορισμούς για το ενισχυμένο στιγμιότυπο του CAP για την απαρίθμηση d_2

4.5.1.5 Επαγόμενοι γράφοι

Σε προηγούμενο εδάφιο, έχουμε αναφέρει ότι η απαρίθμηση d_2 δεν επιφέρει αποδοτική BE υλοποίηση, λόγω της $h^{x_2}(x_1, x_3, x_4, x_5)$, η οποία συνεπάγεται συνένωση τεσσάρων μεταβλητών, κάτι που δε συμβαίνει για την απαρίθμηση d_1. Πώς αυτό αναδεικνύεται στους γράφους με περιορισμούς;

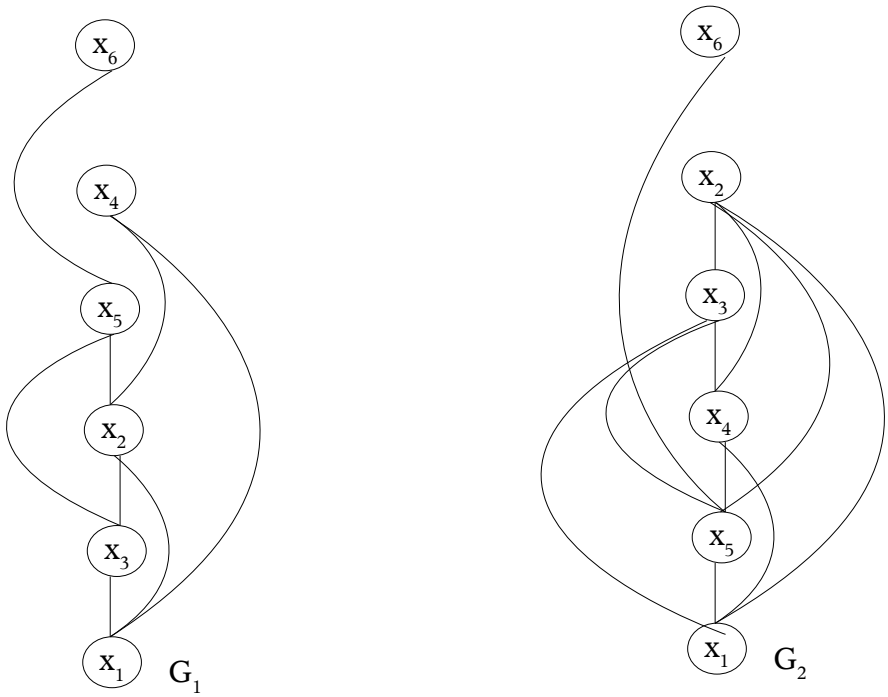
Ας ονομάσουμε G_1 το γράφο με περιορισμούς που παίρνουμε για την απαρίθμηση d_1 και G_2 για τη d_2. Στο παρακάτω σχήμα έχουμε τους G_1 και G_2 δίπλα δίπλα προς σύγκριση.



Σχήμα 6: G_1: Γράφος με περιορισμούς για την απαρίθμηση d_1

G_2: Γράφος με περιορισμούς για την απαρίθμηση d_2

Στο παραπάνω σχήμα, δεν μπορούμε να διακρίνουμε κάτι το οποίο να μπορεί να εξηγεί γιατί στη d_1 η h^{x_2} έχει δύο ορίσματα($h^{x_2}(x_1, x_3)$), ενώ στη d_2 έχει τέσσερα($h^{x_2}(x_1, x_3, x_4, x_5)$). Εύκολα, μπορεί κάποιος να καταλάβει γιατί δε διακρίνεται αυτό. Διότι, και στους δύο γράφους δε φαίνονται οι απαριθμήσεις. Ας ζωγραφίσουμε τους γράφους διαφορετικά, αλλά χωρίς να αλλάξουμε τους κόμβους και τις ακμές του γράφου:



Σχήμα 7: G_1: Γράφος με περιορισμούς για την απαρίθμηση $d_1 = (x_1,x_3,x_2,x_5,x_4,x_6)$

G_2: Γράφος με περιορισμούς για την απαρίθμηση $d_2 = (x_1,x_5,x_4,x_3,x_2,x_6)$

Ας δούμε τι συμβαίνει με τη h^{x_2} . Και στους δύο γράφους, η x_2 έχει τέσσερις προσπίπτουσες ακμές, αλλά στο G_1 η x_2 έχει δύο ακμές «από επάνω» και δύο ακμές «από κάτω», ενώ στο G_2 έχει και τις τέσσερις ακμές «από κάτω». Στο G_1, οι ακμές που έχουν κατεύθυνση προς τα κάτω οδηγούν στις μεταβλητές x_1 και x_3 οι οποίες είναι και στην εμβέλεια της h^{x_2} . Από ότι φαίνεται, σημασία έχει το πόσες ακμές οδηγούν «προς τα κάτω».

Αυτό το χαρακτηριστικό ενός γράφου έχει οριστεί, και στα επόμενα εδάφια θα παραθέσουμε τη εμπλεκόμενη θεωρία.

Για να μπορέσουμε να συγκρίνουμε τους δύο γράφους με περιορισμούς G_1 και G_2 που πήραμε για τις δύο απαριθμήσεις d_1 και d_2, έπρεπε να τους ζωγραφίσουμε με άλλο τρόπο, κατακόρυφα και σύμφωνα με την αντίστοιχη απαρίθμηση. Φαινομενικά στους γράφους δεν άλλαξε κάτι, ούτε οι κόμβοι, ούτε οι ακμές. Άλλαξε κάτι που δεν το απεικονίσαμε στα σχήματα που παραθέσαμε. Οι νέοι γράφοι ήταν κατευθυνόμενοι! Οι ακμές που έρχονταν «από επάνω» και οι ακμές που οδηγούσαν «προς τα κάτω» είναι στην πραγματικότητα κατευθυνόμενες ακμές.

Ορισμός: Διατεταγμένος γράφος

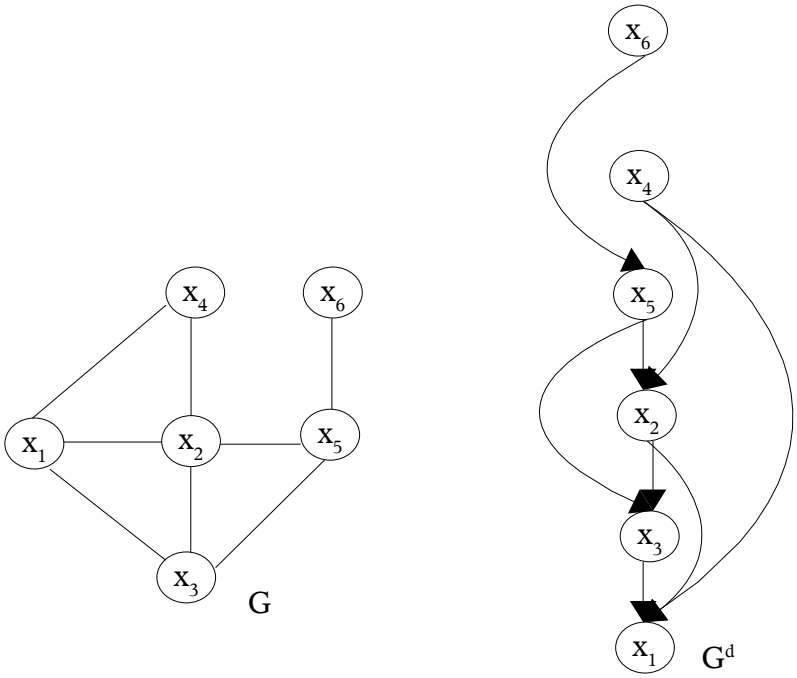
> *Διατεταγμένος γράφος (ordered graph) G^d, ενός μη κατευθυνόμενου γράφου G για μια απαρίθμηση d, είναι ο κατευθυνόμενος γράφος που έχει σαν κόμβους, τους κόμβους του G. Για κάθε ακμή του G, $<x_i,x_j>$, υπάρχει μία ακμή, (x_j,x_i), στο G^d, που έχει κατεύθυνση από τον κόμβο που βρίσκεται πιο δεξιά στη d προς τον κόμβο που βρίσκεται πιο αριστερά. Στο συμβολισμό μας, ο x_j βρίσκεται πιο δεξιά στη d από το x_i. Το διατεταγμένο γράφο τον συμβολίζουμε και με (G,d).*
>
> *Το σύνολο των κόμβων, που έχουν εξερχόμενες ακμές με κατεύθυνση προς ένα κόμβο x_i, το καλούμε παιδιά (children) του x_i. Το σύνολο των κόμβων που*

έχουν εισερχόμενες ακμές από ένα κόμβο x_i, καλείται γονείς (parents) του x_i.

Πλάτος ενός κόμβου (node width) είναι η πληθικότητα των γονέων του. Πλάτος μιας απαρίθμησης d (width of an ordering), w(d), είναι το μέγιστο πλάτος από τους κόμβους που απαριθμούνται στη d. Πλάτος του γράφου (graph width) καλείται το ελάχιστο πλάτος όλων των πιθανών απαριθμήσεων των κόμβων του γράφου.

Παράδειγμα διατεταγμένου γράφου

Δίνουμε ένα παράδειγμα διατεταγμένου γράφου, για γράφο με περιορισμούς που συναντήσαμε στο παράδειγμα για BE. Για απαρίθμηση χρησιμοποιούμε τη d = (x_1,x_3,x_2,x_5,x_4,x_6).



Σχήμα 8: Γράφος G, και ο διατεταγμένος του γράφος G^d, κατά την απαρίθμηση d = (x_1,x_3,x_2,x_5,x_4,x_6)

Παρατηρούμε ότι τα παιδιά του x_2 είναι οι {x_4,x_5} και οι γονείς του οι {x_3,x_1}. Το πλάτος του x_2 είναι δύο, όσο και η πληθικότητα των γονέων του. Το πλάτος της d είναι και αυτό δύο, αφού το πλάτος του x_6 είναι ένα, του x_4 δύο, του x_5 δύο, του x_2 δύο, του x_2 ένα και του x_1 μηδέν.

Πόσο είναι όμως το πλάτος του G; Είναι ίσο με το ελάχιστο πλάτος όλων των πιθανών απαριθμήσεων των κόμβων του. Αφού η απαρίθμηση d του G δίνει πλάτος δύο, τότε ξέρουμε ότι το πλάτος του G είναι μικρότερο ή ίσο του δύο. Θα δούμε θεώρημα που λέει ότι εάν ένας γράφος έχει πλάτος ένα, τότε είναι δέντρο. Ο G δεν είναι δέντρο, άρα δεν μπορεί να έχει πλάτος ένα. Άρα, ο G έχει πλάτος δύο.

Θεώρημα: Πλάτος δέντρου

Εάν το πλάτος ενός γράφου είναι ένα, τότε αυτός είναι δέντρο.

Απόδειξη:

Η απόδειξη καθώς και το θεώρημα είναι απλά. Αφού το πλάτος του γράφου

είναι δέντρο, τότε υπάρχει απαρίθμηση για την οποία ο διατεταγμένος γράφος έχει την εξής ιδιότητα: όλοι οι κόμβοι έχουν πλάτος ίσο με ένα. Αυτό συνεπάγεται ότι όλοι οι κόμβοι έχουν ένα μόνο πατέρα, ιδιότητα αρκετή να ορίσει ένα δέντρο.

Γυρνώντας στην προσπάθεια να ορίσουμε το μέγεθος που θα εκφράζει το πόσο καλή είναι μία απαρίθμηση για ΒΕ, καταλήγουμε στο ότι μας ενδιαφέρει το πλάτος της απαρίθμησης για το γράφο με περιορισμούς του προβλήματος. Όχι ακριβώς! Σκεφτείτε την απαρίθμηση d_2. Αυτή αλλάζει το γράφο με περιορισμούς σε ένα καινούργιο γράφο με περιορισμούς προσθέτοντας ακμές. Επομένως, δε πρέπει να υπολογίσουμε το πλάτος της d_2 για τον αρχικό γράφο με περιορισμούς, αλλά για τον τελικό, που προκύπτει με το πέρας των ΒΕ.

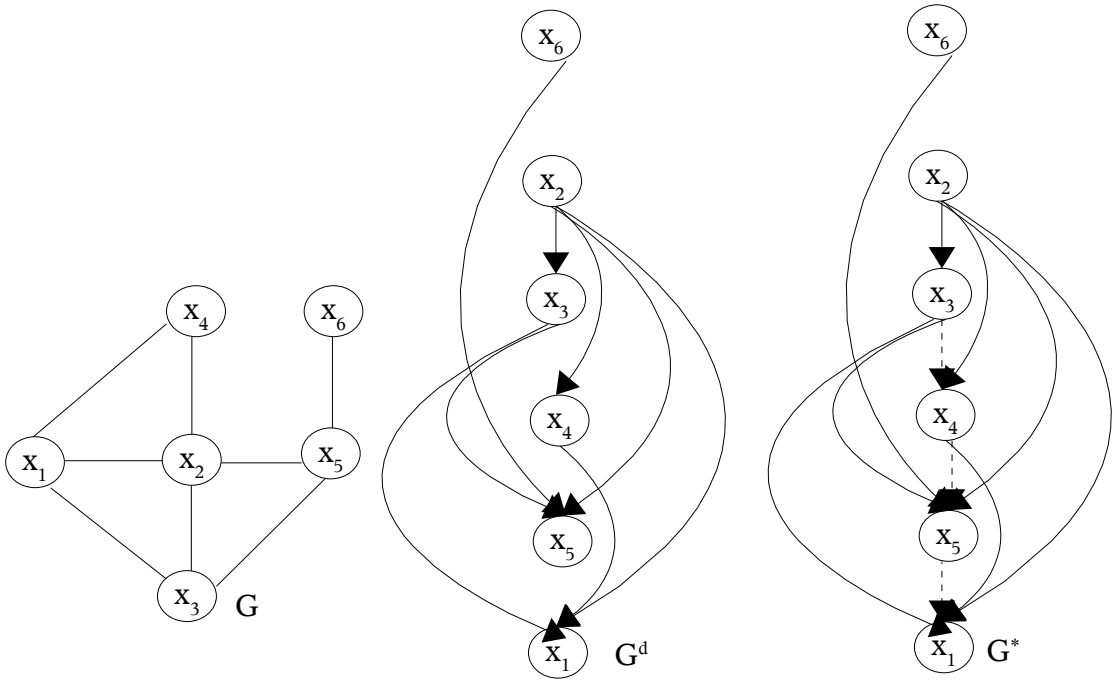
Μπορούμε να γνωρίζουμε πριν ξεκινήσουμε την εκτέλεση ενός ΒΕ, το τελικό γράφο; Η απάντηση είναι ναι! Θα δώσουμε ένα ορισμό που θα βοηθήσει στην εύρεση του τρόπου με τον οποίο θα υπολογίσουμε τον τελικό γράφο.

Ορισμός: Επαγόμενος γράφος

Ο επαγόμενος γράφος (induced graph) G^* ενός διατεταγμένου γράφου (G,d) είναι ένας διατεταγμένος γράφος (G,d). Ο G* έχει σαν κόμβους τους κόμβους του G. Σαν ακμές έχει τις ακμές του (G,d) συν τις ακμές που προκύπτουν, εάν για κάθε κόμβο στο (G,d) ενώσουμε μεταξύ τους, τους γονείς του.*

Το επαγόμενο πλάτος $w^*(d)$ ενός διατεταγμένου γράφου (induced width of an ordered graph) (G,d) είναι το πλάτος του επαγόμενου γράφου (G^*,d). Το επαγόμενο πλάτος (induced width of a graph) w^* ενός γράφου G, είναι το ελάχιστο επαγόμενο πλάτος από όλους τους δυνατούς διατεταγμένους γράφους του G.

Παραθέτουμε ένα παράδειγμα με επαγόμενο γράφο.



Σχήμα 9: Γράφος G, ο διατεταγμένος του γράφος (G,d) και ο επαγόμενος γράφος (G^*,d), κατά την απαρίθμηση d = (x_1,x_5,x_4,x_3,x_2,x_6). Με διακεκομμένη αναπαριστώνται οι ακμές που προστέθηκαν κατά τη δημιουργία του G^*.

Προσθέσαμε τις ακμές (x_3,x_4), (x_4,x_5) και (x_5,x_1). Οι γράφοι αυτοί, είναι οι γράφοι του ενισχυμένου παραδείγματος CAP που είδαμε με τη d_2 απαρίθμηση. Ο G είναι ο γράφος με περιορισμούς. Ο G^* είναι ο G, όπως αυτός προκύπτει μετά από BE και συνάμα είναι ο επαγόμενος γράφος του G. Υπάρχει μία μεγάλη υπόνοια ότι ο επαγόμενος γράφος είναι ο γράφος που παίρνουμε μετά από BE. Είναι αλήθεια!

Ας μελετήσουμε τι συμβαίνει κατά τους BE. Τους κόμβους μεταβλητές, τους εξετάζουμε με την αντίστροφη σειρά της απαρίθμησης d_2. Τους εξετάζουμε όπως είναι στο διατεταγμένο γράφο, ξεκινώντας από τον τελευταίο κόμβο της d_2 και πηγαίνοντας προς τα μπροστά. Στους BE, κάθε φορά που τελειώνουμε με έναν κόμβο x_i, εισάγουμε στο δίκτυο έναν ασθενή περιορισμό h^i. Η εμβέλεια αυτού του h^i συμπεριλαμβάνει τις μεταβλητές που έχουν την εξής ιδιότητα: ανήκουν μαζί με το x_i στην εμβέλεια κάποιου περιορισμού και είναι στην απαρίθμηση πιο αριστερά από το x_i, σε γραφοθεωρητικά λόγια, είναι γονείς του x_i. Όλες αυτές οι μεταβλητές – γονείς ανήκουν στον ίδιο ασθενή περιορισμό και επομένως πρέπει να ενωθούν αναμεταξύ τους και αυτές με ακμές. Ενώνοντας τους γονείς του x_i αναμεταξύ τους με ακμές, παίρνουμε τον επαγόμενο γράφο!

Για BE αλγορίθμους, μία απαρίθμηση είναι η καλύτερη εάν το επαγόμενο πλάτος της είναι ίσο με το επαγόμενο πλάτος του γράφου. Στο παράδειγμά μας, η απαρίθμησή d_1 έχει επαγόμενο πλάτος δύο και το επαγόμενο πλάτος του γράφου με περιορισμούς είναι δύο. Δεν υπάρχει επομένως απαρίθμηση d_i που να συμπεριφέρεται γνησίως καλύτερα από τη d_1 για BE. Δυστυχώς, το να βρεθεί για ένα γράφο μία απαρίθμηση που το επαγόμενο πλάτος της να είναι ίσο με το επαγόμενο πλάτος του γράφου ανήκει στο NP! Οπότε, δεν μπορούμε να ξέρουμε πόσο καλή μπορεί να είναι μια απαρίθμησή.

4.6 Βιβλιογραφικές αναφορές

Πολλά προβλήματα μπορούν να μοντελοποιηθούν σαν COP. Ένα πολύ γνωστό, είναι ο

κανόνας του Golomb (the Golomb ruler). Η μοντελοποίηση του συγκεκριμένου προβλήματος είναι πολύ ενδιαφέρουσα [SSW99].

Μία πάρα πολλή καλή αναφορά στη απαραίτητη γραφοθεωρία της εργασίας αυτής, κάνουν οι [Rin03] και [Sch99], συμπληρώνοντας ο ένας τον άλλο.

Μια προσέγγιση στους VE γίνεται στο [Shc08]. Για την πολυπλοκότητα των BE, μπορείτε να δείτε για περισσότερες πληροφορίες [Lar01].

5 Απαλοιφή με μικρο-κάδους

5.1 Η «αδυναμία» του ELIM-OPT-CONS

Έχουμε πει ότι η πολυπλοκότητα της ELIM-OPT-CONS είναι $O(r \cdot k^{w^*(d)})$, όπου k είναι το μέγιστο από τα μεγέθη των πεδίων τιμών των μεταβλητών, r είναι το πλήθος των ασθενών μεταβλητών και $w^*(d)$ είναι το επαγόμενο πλάτος της απαρίθμησης που επιλέχθηκε. Η πολυπλοκότητα είναι εκθετική στο επαγόμενο πλάτος και η χρησιμοποίηση του ELIM-OPT-CONS κρίνεται ακατάλληλη για πραγματικά συστήματα.

5.1.1 Μικρο-κάδοι

Ένας αλγόριθμος που βασίζεται στον ELIM-OPT-CONS και είναι BE, είναι ο αλγόριθμος απαλοιφής με μικρο-κάδους (mini-bucket elimination algorithm). Συχνά, θα καλούμε τον αλγόριθμο απαλοιφής με μικρο-κάδους σαν MBE από το ακρωνύμιο που σχηματίζεται από τις λέξεις mini-bucket elimination.

Για να μπούμε στο ζουμί του MBE, θα δούμε το κλασσικό μας παράδειγμα με τη δεύτερη απαρίθμηση d_2. Θέλουμε να ελαττώσουμε την πολυπλοκότητα του υπολογισμού αυτής της συνάρτησης:

$$M = \max_{x_1}\{F_1(x_1) + \max_{x_5}\{F_5(x_5) + \max_{x_4}\{F_4(x_4) + \max_{x_3}\{F_3(x_3) + F_{13}(x_1, x_3) +$$

$$+ \max_{x_2}\{F_2(x_2) + F_{12}(x_1, x_2) + F_{235}(x_2, x_3, x_5) + F_{124}(x_1, x_2, x_4) + \max_{x_6}\{F_6(x_6) + F_{56}(x_5, x_6)\}\}\}\}\}$$

Το πρόβλημά μας είναι, ότι για την απαρίθμηση d_2 το επαγόμενο πλάτος είναι τέσσερα και αυτό σημαίνει ότι κάποια συνάρτηση που συμμετέχει στη συνάρτηση κόστος θα έχει σαν όρισμα τέσσερις μεταβλητές. Συγκεκριμένα,

$$h^{x_2}(x_1, x_3, x_4, x_5) = \max_{x_2}\{F_2(x_2) + F_{12}(x_1, x_2) + F_{235}(x_2, x_3, x_5) + F_{124}(x_1, x_2, x_4)\}$$

Μπορεί αυτό να αλλάξει; Γνωρίζουμε ότι εάν έχουμε δύο συναρτήσεις α και β με πεδίο τιμών το R^+, οι θετικοί πραγματικοί αριθμοί, τότε $\max(\alpha+\beta) \leqslant \max(\alpha) + \max(\beta)$. Μπορούμε να γράψουμε:

$$h^{x_2}(x_1, x_3, x_4, x_5) \leqslant \max_{x_2}\{F_2(x_2) + F_{12}(x_1, x_2) + F_{124}(x_1, x_2, x_4)\} + \max_{x_2}\{F_{235}(x_2, x_3, x_5)\}$$

$$h^{x_2}(x_1, x_3, x_4, x_5) \leqslant h^{x_2}(x_1, x_4) + h^{x_2}(x_3, x_5)$$

Αποφεύγουμε τη συνένωση τεσσάρων μεταβλητών (στην πραγματικότητα πέντε) και χρειάζεται μόνο να υπολογίσουμε συναρτήσεις δύο μεταβλητών (στην πραγματικότητα τριών). Δυστυχώς, χάνουμε σε ακρίβεια, καθώς η συνάρτηση θα υπερεκτιμηθεί από το άθροισμα $h^{x_2}(x_1, x_4) + h^{x_2}(x_3, x_5)$. Πόσο ακριβώς χάνουμε σε ακρίβεια; Η απάντηση είναι δεν ξέρουμε καθώς η απόκλιση εξαρτάται από τη φύση των συναρτήσεων που συμμετέχουν.

Εντούτοις, η απόκλιση εξαρτάται και από το σε πόσα κομμάτια διαμελίζουμε τη συνάρτηση. Τη $h^{x_2}(x_1, x_3, x_4, x_5)$ τη χωρίσαμε σε δύο κομμάτια, $h^{x_2}(x_1, x_4)$ και $h^{x_2}(x_3, x_5)$. Σε όσο πιο πολλά κομμάτια τη διαμελίσουμε τη $h^{x_2}(x_1, x_3, x_4, x_5)$, τόσο πιο ανακριβής θα είναι η εκτίμηση που θα πάρουμε για αυτήν.

Για να χωρίσουμε τη $h^{x_2}(x_1, x_3, x_4, x_5)$ διαφορετικά:

$$h^{x_2}(x_1,x_3,x_4,x_5) \leqslant \max_{x_2}\{F_2(x_2)+F_{12}(x_1,x_2)\}+\max_{x_2}\{F_{235}(x_2,x_3,x_5)+F_{124}(x_1,x_2,x_4)\}$$

$$h^{x_2}(x_1,x_3,x_4,x_5) \leqslant h^{x_2}(x_1)+h_{mbe}{}^{x_2}(x_1,x_3,x_4,x_5)$$ [4]

Ο παραπάνω διαχωρισμός της $h^{x_2}(x_1,x_3,x_4,x_5)$ σε $h^{x_2}(x_1)$ και $h_{mbe}{}^{x_2}(x_1,x_3,x_4,x_5)$ δεν έχει αποφέρει το αποτέλεσμα που θα θέλαμε. Η $h_{mbe}{}^{x_2}(x_1,x_3,x_4,x_5)$ έχει τέσσερα ορίσματα πάλι. Αυτή η παρατήρηση μας οδηγεί στο να μην είναι κριτήριο το σε πόσα τεμάχια θα κόψουμε τη $h^{x_2}(x_1,x_3,x_4,x_5)$, αλλά σε πόσες μεταβλητές θα είναι ορισμένες οι νέες ανακύπτουσες συναρτήσεις! Στο προηγούμενο παράδειγμα, οι καινούργιες συναρτήσεις είχαν δύο μεταβλητές και αυτό μας άρεσε πολύ!

Σε κάθε κάδο έχουμε συναρτήσεις που θέλουμε να τις χωρίσουμε ανάλογα με τις μεταβλητές τους. Σας θυμίζει κάτι αυτό; Οι ίδιοι οι κάδοι είναι κατασκευασμένοι με αυτό το σκεπτικό. Κάθε κάδος φιλοξενεί συναρτήσεις, που το πιο δεξιά στην απαρίθμηση όρισμά τους είναι η μεταβλητή που αντιστοιχεί ο κάδος. Το ίδιο θα κάνουμε και για τις νέες συναρτήσεις που πήραμε διασπώντας τη συνάρτηση του κάδου. Θα δημιουργήσουμε νέους μικρότερους κάδους (μικρο-κάδους) και θα τοποθετούμε εκεί τις νέες συναρτήσεις σύμφωνα με το κριτήριο της δεξιότερης στην απαρίθμηση μεταβλητής.

Για το χωρισμό της $h^{x_2}(x_1,x_3,x_4,x_5)$ σε $h^{x_2}(x_1,x_4)$ και $h^{x_2}(x_3,x_5)$ θα φτιάξουμε δύο μικρο-κάδους, ένα για τη $h^{x_2}(x_1,x_4)$ και θα αντιστοιχίζεται στη μεταβλητή x_4 και ένα για τη $h^{x_2}(x_3,x_5)$ που θα αναφέρεται στη x_3. Θυμίζουμε, ότι ακολουθούμε την απαρίθμηση $d_2 = (x_1,x_5,x_4,x_3,x_2,x_6)$.

Είναι πολύ σημαντικό να εξηγήσουμε γιατί $h^{x_2}(x_1,x_3,x_4,x_5) \leqslant h^{x_2}(x_1,x_4)+h^{x_2}(x_3,x_5)$. Δεν είναι τόσο απλό όσο φαίνεται το πρόβλημα. Οι συναρτήσεις, αρκετά συχνά, μπορεί να είναι απλές συναρτήσεις για τις οποίες να ισχύει $h^{x_2}(x_1,x_3,x_4,x_5)=h^{x_2}(x_1,x_4)+h^{x_2}(x_3,x_5)$. Θέλει μεγάλη προσοχή γιατί μπορεί να εξαπατηθούμε και να πέσουμε στην παραπάνω παγίδα. Έστω ότι $h^{x_2}(x_1,x_3,x_4,x_5)=\max_{x_2}(2x_2+x_1+x_3+x_4+x_5)$ και ότι τη χωρίζουμε σε $h^{x_2}(x_1,x_4)=\max_{x_2}(x_2+x_1+x_4)$ και $h^{x_2}(x_3,x_5)=\max_{x_2}(x_2+x_3+x_5)$. Εάν το πεδίο τιμών όλων των μεταβλητών είναι οι θετικοί αριθμοί, τότε $h^{x_2}(x_1,x_3,x_4,x_5)=h^{x_2}(x_1,x_4)+h^{x_2}(x_3,x_5)$. Λάθος! Που είναι το λάθος; Μπορεί να υπάρχει ένας σκληρός περιορισμός που να απαγορεύει την αποτίμηση $x_1 = x_2 = x_3 = x_4 = x_5$. Αυτός ο περιορισμός θα πάει παρέα με μία από τις $h^{x_2}(x_1,x_4)$ και $h^{x_2}(x_3,x_5)$ στον ίδιο μικρο-κάδο. Επομένως, μία από τις δύο νέες συναρτήσεις δε θα γνωρίζει την ύπαρξη του περιορισμού και θα ορίζεται σε παραπάνω πλειάδες. Οπότε, $h^{x_2}(x_1,x_3,x_4,x_5)\neq h^{x_2}(x_1,x_4)+h^{x_2}(x_3,x_5)$. Το παραπάνω σενάριο δε συμβαίνει πάντα, αλλά είναι το πιο πιθανό!

5.2 Αλγόριθμος MBE-OPT-CONS(i)

Εξαιτίας της ομοιότητας στην κατασκευή των μικρο-κάδων με τους κάδους, ο MBE είναι παρόμοιος με τον BE.

Στον MBE ελέγχουμε το πλήθος των ορισμάτων των νέων συναρτήσεων. Αυτό θα περνάει σαν παράμετρο στον αλγόριθμο μας, μέσω ενός θετικού ακεραίου i.

Ο MBE υπολογίζει ένα διάστημα όπου κυμαίνεται η τιμή της συνάρτησης κόστους. Το δεξιό άκρο του διαστήματος είναι αυτό που συζητήσαμε και αναλύσαμε. Το αριστερό είναι αυτό που θα υπολογίζεται από την αποτίμηση που θα επιστρέψει ο αλγόριθμος και το οποίο θα είναι μικρότερο

4 Ονομάζουμε τη δεύτερη δημιουργηθείσα συνάρτηση $h_{mbe}{}^{x_2}$ γιατί αν την ονομάζαμε h^{x_2} θα υπήρχε σύγχυση μεταξύ αυτής και της αρχικής h^{x_2}.

ή ίσο με το βέλτιστο. Δε γίνεται να είναι μεγαλύτερο γιατί τότε μια αποτίμηση θα απόφερε μεγαλύτερο κέρδος από το βέλτιστο που είναι άτοπο.

Ορίστε ο MBE-OPT-CONS(i):

Αλγόριθμος MBE-OPT-CONS(i)

Δεδομένα:

Ένα δίκτυο με περιορισμούς (X,D,C_h,C_s), με $C_h=\{R_{S_1},\dots,R_{S_l}\}, C_s=\{F_{Q_1},\dots,F_{Q_m}\}$, l = |C_h| και m = |C_s|.

Μία απαρίθμηση d των μεταβλητών X.

Το πλήθος των ορισμάτων i των νέων συναρτήσεων στους μικρο-κάδους.

Αρχή Αλγορίθμου:

Για κάθε ισχυρό περιορισμό R_{S_z} , με z από το 1 στο l, **κάνε:**

S_z = η εμβέλεια του R_{S_i}

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη S_z

Τοποθέτησε τον R_{S_z} στον κάδο Bucket(x_b)

Τέλος Για κάθε

Για κάθε ασθενή περιορισμό F_{Q_z} , με z από το 1 στο m, **κάνε:**

Q_z = η εμβέλεια της F_{Q_z}

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη Q_z

Τοποθέτησε τη F_{Q_z} στον κάδο Bucket(x_b)

Τέλος Για κάθε

Για κάθε κάδο Bucket(x_p), με p από το n στο 1, **κάνε:**

Διαμέρισε τους ισχυρούς περιορισμούς, που υπάρχουν στο Bucket(x_p), σε μικρο-κάδους, έτσι ώστε κάθε κάδος να περιέχει περιορισμούς που η εμβέλεια τους είναι υποσύνολο i μεταβλητών.

Διαμέρισε τις συναρτήσεις-περιορισμούς, που έχουν προστεθεί στο Bucket(x_p), σε μικρο-κάδους, έτσι ώστε κάθε κάδος να περιέχει συναρτήσεις που η εμβέλεια τους είναι υποσύνολο i μεταβλητών.

Διαμέρισε τους αρχικούς ασθενείς περιορισμούς, που υπάρχουν στο Bucket(x_p), σε μικρο-κάδους, έτσι ώστε κάθε κάδος να περιέχει περιορισμούς που η εμβέλεια τους είναι υποσύνολο i μεταβλητών.

Για κάθε μικρο-κάδο mb **κάνε:**

$R_{S_1},\dots,R_{S_k}$ = οι ισχυροί περιορισμοί που υπάρχουν στο mb

$S_1,\dots,S_k$ = οι εμβέλειες των $R_{S_1},\dots,R_{S_k}$

U_{mb} = ($\bigcup_i S_i$) – {x_{mb}}

$R^{mb}=\pi_{U_{mb}}$ JOINT $\prod_{i=1}^{k} R_i$,ο καθολικός περιορισμός του mb

$h^{x_1},\dots,h^{x_j}$ = οι προστιθέμενες συναρτήσεις-περιορισμοί στο mb

$Q_1,\dots,Q_j$ = οι εμβέλειες των $h^{x_1},\dots,h^{x_j}$

$F_{Q_1},\dots,F_{Q_o}$ = οι αρχικοί ασθενείς περιορισμοί στο mb

$Q_1,\dots,Q_o$ = οι εμβέλειες των $F_{Q_1},\dots,F_{Q_o}$

V_{mb} = ($\bigcup_i Q_i$) – {τις μεταβλητές που αντιπροσωπεύει ο mb}

$W_{mb} = U_{mb} \bigcup V_{mb}$

Για κάθε πλειάδα t του W_{mb} **κάνε:**

$$h^{mb}(t)=\max_{a_p \text{ έ.ώ. } R^{x_p}(t,a_p)} \{\sum_{i=1}^{j} h^{x_i}(t,a_p)+\sum_{i=1} F_{Q_i}(t,a_p)\}$$

x_b = η πιο δεξιά μεταβλητή στη d που ανήκει στο W_{mb}

Τοποθέτησε τη h^{x_p} στο Bucket(x_b)

x_g = η πιο δεξιά μεταβλητή στην d που ανήκει στο U_{mb}
Τοποθέτησε τη R^{mb} στο Bucket(x_g)

Τέλος Για κάθε

Τέλος Για κάθε

Τέλος Για κάθε

right_bound = το κόστος που υπολογίζεται στο Bucket(x_1).

Για i από το 1 στο n, **κάνε**

Δεδομένης αποτίμησης $\vec{a_{i-1}}$, αποτίμησε το x_i με μια τιμή a_i ε D_i, που να μεγιστοποιεί το άθροισμα των συναρτήσεων στο Bucket(x_i)

Τέλος Για

left_bound = το κόστος που προκύπτει από την αποτίμηση $\vec{a_n}$

Τέλος Αλγορίθμου

Έξοδος:

Μια συνεπής αποτίμηση που αποφέρει ένα διάστημα για την τιμή του μέγιστου κόστους $\sum_{F_i \in C_s} F_i$. Το διάστημα αυτό είναι το [left_bound,right_bound].

Πριν συνεχίσουμε με την πολυπλοκότητα του αλγορίθμου, ας βεβαιωθούμε ότι ο MBE-OPT-CONS(i) υπολογίζει αυτά που ισχυρίζεται.

Θεώρημα: Ορθότητα MBE-OPT-CONS(i)

Ο MBE-OPT-CONS(i) βρίσκει ένα διάστημα τιμών για τη συνάρτηση κόστους.

Απόδειξη:

Για το δεξιό άκρο του διαστήματος:

με τη διάσπαση των κάδων σε μικρο-κάδους δε γίνεται συμπερασμός (inference) μεταξύ όλων των μεταβλητών του κάδου. Οι μικροί κάδοι δεν αποκλείουν διάφορες αποτιμήσεις, που είναι συνεπείς εκείνη τη στιγμή, αλλά που είναι αργότερα ασυνεπείς. Αποτέλεσμα είναι οι μικρο-κάδοι να υπερεκτιμούν τη συνάρτηση κόστους. Δυστυχώς, αυτήν την υπερεκτίμηση τη μεταδίδουν και στους επόμενους κάδους. Ο αλγόριθμος υπερεκτιμά τη συνάρτηση κόστους και το δεξιό άκρο είναι πιο μεγάλο ή ίσο με αυτή.

Όσον αφορά το αριστερό άκρο:

αυτό υπολογίζεται από την αποτίμηση που επιστρέφει ο ELIM-OPT-CONS. Αυτό είναι μικρότερο ή ίσο από τη βέλτιστη τιμή της συνάρτησης κόστους. Έστω ότι δεν είναι, τότε η αποτίμηση βρίσκει σαν τιμή της συνάρτησης κόστους μια τιμή M, που είναι μεγαλύτερη από τη βέλτιστη τιμή. Άτοπο, γιατί η M θα ήταν η βέλτιστη τιμή.

5.2.1 i – διαμέριση

Μένει να σχολιάσουμε το εξής: στον MBE-OPT-CONS(i) δεν αναφέρουμε με ποιον αλγόριθμο γίνεται η διαμέριση (partitioning) των κάδων σε μικρο-κάδους. Με ποιο κριτήριο

επιλέγουμε τους συγκεκριμένους μικρο-κάδους.

Για $h^{x_2}(x_1, x_3, x_4, x_5) = \max_{x_2}\{F_2(x_2) + F_{12}(x_1, x_2) + F_{235}(x_2, x_3, x_5) + F_{124}(x_1, x_2, x_4)\}$ και i=3 έχουμε, τουλάχιστον αυτές τις δύο, εναλλακτικές προτάσεις.

- Να χωρίσουμε τη $h^{x_2}(x_1, x_3, x_4, x_5)$ σε $h^{x_2}(x_3, x_5) = \max_{x_2}\{F_2(x_2) + F_{235}(x_2, x_3, x_5)\}$ και $h^{x_2}(x_1, x_4) = \max_{x_2}\{F_{12}(x_1, x_2) + F_{124}(x_1, x_2, x_4)\}$.
- Ή χωρίζουμε σε $h^{x_2}(x_3, x_5) = \max_{x_2}\{F_{235}(x_2, x_3, x_5)\}$, $h^{x_2}(x_1) = \max_{x_2}\{F_2(x_2) + F_{12}(x_1, x_2)\}$ και $h^{x_2}(x_1, x_4) = \max_{x_2}\{F_{124}(x_1, x_2, x_4)\}$

Το κριτήριο που θα επιλέξουμε τη διαμέριση, είναι δύσκολο να αποδειχθεί ότι θα είναι και το καλύτερο. Αν και στο προηγούμενο παράδειγμα η πρώτη διαμέριση προφανώς υπερτερεί της δεύτερης, καθώς έχουμε λιγότερους μικρο-κάδους. Επί το πλείστον, μια τέτοια διαμέριση, που μας αποφέρει το μικρότερο αριθμό μικρο-κάδων, προτιμάται από τις άλλες, αλλά χωρίς να είναι βέβαιο ότι είναι η καλύτερη.

Στον MBE-OPT-CONS(i), δεν παραθέσαμε μέθοδο διαμέρισης σε συναρτήσεις των i ορισμάτων (i – διαμέριση) και αφήνεται στον αναγνώστη να υλοποιήσει το δικό του τρόπο i – διαμέρισης (i – partitioning).

5.2.2 Πολυπλοκότητα MBE-OPT-CONS(i)

Αρχικά, θα περιμέναμε ότι η πολυπλοκότητα του MBE-OPT-CONS(i), θα είναι περίπου ίδια με του ELIM-OPT-CONS. Αυτό αποκλείεται για δύο λόγους. Πρώτον, η ανάγκη της εφεύρεσης αυτού του αλγορίθμου είναι η εύρεση ενός με καλύτερη πολυπλοκότητα από τον ELIM-OPT-CONS. Δεύτερον, οι συναρτήσεις έχουν το πολύ i μεταβλητές σαν όρισμα, και το i δεν έχει καμία σχέση με το επαγόμενο πλάτος $w^*(d)$ της απαρίθμησης d που χρησιμοποιείται.

Πριν μελετήσουμε την πολυπλοκότητα του MBE πρέπει να αναφέρουμε ότι αν τρέξουμε τον MBE-OPT-CONS(i) με $i \geqslant w^*(d)$, τότε ο MBE-OPT-CONS(i) και ο ELIM-OPT-CONS είναι ίδιοι! Αυτό επιβεβαιώνεται και από τα παρακάτω θεωρήματα:

Θεώρημα: Πληρότητα MBE-OPT-CONS(i)

> *Ο MBE-OPT-CONS(i) είναι πλήρης, εάν για την απαρίθμηση d που εκτελείται, ισχύει $w^*(d) \leqslant i$.*

Απόδειξη:

> *Προφανώς, εάν $w^*(d) \leqslant i$, τότε κάθε μικρο-κάδος είναι κανονικός κάδος αφού για τη μεταβλητή x_i που φιλοξενεί θα εξετάσει όλους τους γονείς της, $parents(x_i)$ μαζί.*

Θεώρημα: Πολυπλοκότητα MBE-OPT-CONS(i)

> *Η πολυπλοκότητα του MBE-OPT-CONS(i), σε χρόνο και χώρο, είναι $O(r \cdot exp(i))$, όπου r είναι ο συνολικός αριθμός των ασθενών περιορισμών και i η παράμετρος που καθορίζει το μέγεθος των μικρο-κάδων.*

Απόδειξη:

> *Η πολυπλοκότητα αποφασίζεται από τους υπολογισμούς των συναρτήσεων.*

Επομένως, η πολυπλοκότητα ισούται με τους πόρους που σπαταλώνται για κάθε συνάρτηση επί το πλήθος όλων των συναρτήσεων συμπεριλαμβανομένων αυτών που γεννιούνται κατά τη διάρκεια του αλγορίθμου.

Κάθε φορά που διασπάται μια h^{x_1} *και γίνονται πράξεις στους αντίστοιχους μικρο-κάδους, εξαλείφεται μία μεταβλητή από την εμβέλεια του* h^{x_1} *. Μπορούμε να φανταστούμε την όλη διαδικασία σαν ένα δέντρο. Στα φύλλα του δέντρου υπάρχουν οι αρχικές συναρτήσεις – περιορισμοί. Στους εσωτερικούς κόμβους του δέντρου έχουμε συναρτήσεις που έχουν δημιουργηθεί από τα παιδιά τους με απαλοιφή μεταβλητών. Έτσι, η ρίζα αυτού του δέντρου είναι το προσεγγισμένο βέλτιστο κόστος του κάδου Bucket(*x_1*).*

Στα δέντρα, οι εσωτερικοί κόμβοι είναι το πολύ όσα τα φύλλα του. Τα φύλλα εδώ είναι το πλήθος r και επομένως, όλο το δέντρο έχει το πολύ 2·r κόμβους – συναρτήσεις. Ο χρόνος και ο χώρος να υπολογιστεί μια συνάρτηση με i το πλήθος ορίσματα, είναι εκθετικός του i. Οπότε συνολικά έχουμε O(r·exp(i)) πολυπλοκότητα.

Στη βιβλιογραφία, το παραπάνω δέντρο της απόδειξης, μπορούμε να το βρούμε και σαν δέντρο υπολογισμού. Για να γίνει πιο κατανοητό πως σχηματίζεται αυτό το δέντρο, σε επόμενο κεφάλαιο, για το παράδειγμα για MBE-OPT-CONS(i) που ακολουθεί, θα παραθέσουμε το δέντρο υπολογισμού του.

5.2.3 Παράδειγμα MBE-OPT-CONS(i)

Θα δούμε ένα παράδειγμα MBE-OPT-CONS(i) για το γνωστό μας πρόβλημα. Για λόγους ευκολίας, θα παραθέσουμε το πρόβλημά μας εδώ:

- ➔ $X = \{x_1,x_2,x_3,x_4,x_5,x_6\}$, όπου το x_i συμβολίζει τη δημοπρασία b_i
- ➔ $D = \{D_1,D_2,D_3,D_4,D_5,D_6\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = \{0,1\}$.
- ➔ Ασθενείς περιορισμοί C_s:
  - $F_1(x_1)=\begin{cases}8, \text{εάν } x_1=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_2(x_2)=\begin{cases}6, \text{εάν } x_2=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_3(x_3)=\begin{cases}5, \text{εάν } x_3=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_4(x_4)=\begin{cases}2, \text{εάν } x_4=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_5(x_5)=\begin{cases}2, \text{εάν } x_5=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_6(x_6)=\begin{cases}4, \text{εάν } x_6=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_{12}(x_1,x_2)=\begin{cases}10, \text{εάν } x_1=x_2=1\\ 8, \text{εάν } x_1=x_2=0\\ 0, \text{αλλιώς}\end{cases}$

- $F_{13}(x_1,x_3)=\begin{Bmatrix} 8, \text{εάν } x_1=x_3=1 \\ 2, \text{εάν } x_1=1 \text{ και } x_3=0 \\ 10, \text{εάν } x_1=0 \text{ και } x_3=0 \\ 0, \text{αλλιώς} \end{Bmatrix}$
- $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix} 8, \text{εάν } x_2=x_3=x_5 \\ 0, \text{αλλιώς} \end{Bmatrix}$
- $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix} 8, \text{εάν } x_1=x_2=x_4=1 \\ 0, \text{αλλιώς} \end{Bmatrix}$
- $F_{56}(x_5,x_6)=\begin{Bmatrix} 10, \text{εάν } x_5=x_6=1 \\ 6, \text{εάν } x_5=x_6=0 \\ 0, \text{αλλιώς} \end{Bmatrix}$

➔ Ισχυροί περιορισμοί, $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}=R_{13}=R_{14}=R_{24}=R_{25}=R_{35}=\{(1,0),(0,1),(0,0)\}$.

Η απαρίθμηση, που θα χρησιμοποιήσουμε, είναι η $d_2 = (x_6,x_1,x_5,x_2,x_3,x_4)$ και i = 2. Ξεκινάμε, όπως στον ELIM-OPT-CONS, να βάλουμε τους ισχυρούς και ασθενείς περιορισμούς του προβλήματος σε κάδους.

Κάδος	Συναρτήσεις
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $R_{24}(x_2,x_4)$, $R_{14}(x_1,x_4)$
Bucket(x_3):	$F_3(x_3)$, $F_{235}(x_2,x_3,x_5)$, $F_{13}(x_1,x_3)$, $R_{35}(x_3,x_5)$, $R_{13}(x_1,x_3)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $R_{12}(x_1,x_2)$, $R_{25}(x_2,x_5)$
Bucket(x_5):	$F_5(x_5)$, $F_{56}(x_5,x_6)$
Bucket(x_1):	$F_1(x_1)$
Bucket(x_6):	$F_6(x_6)$

Πίνακας 8: Οι κάδοι μετά την αρχικοποίηση του MBE-OPT-CONS(i)

Η αρχικοποίηση των κάδων έχει τελειώσει και συνεχίζουμε με τον κάδο της τελευταίας μεταβλητής της d_2.

Στο Bucket(x_4), θα σχηματισθούν οι mini-bucket(x_1,x_4) και mini-bucket(x_2,x_4).

- Στο mini-bucket(x_1,x_4):
  - Έχουμε τον $R_{14}(x_1,x_4)$ μόνο του. Θα υπολογισθεί ο $R^{x_4}(x_1)=\{0,1\}$, αλλά αυτός είναι καθολικός και δε θα τοποθετηθεί σε επόμενο κάδο.
- Στο mini-bucket(x_2,x_4):
  - Έχουμε τον $R_{24}(x_2,x_4)$ μόνο του. Θα υπολογισθεί ο $R^{x_4}(x_2)=\{0,1\}$, αλλά αυτός είναι καθολικός και δε θα τοποθετηθεί σε επόμενο κάδο.
  - Έχουμε τις $F_4(x_4)$ και $F_{124}(x_1,x_2,x_4)$. Προσοχή, η $F_{124}(x_1,x_2,x_4)$ έχει τρεις μεταβλητές, αλλά δεν μπορεί να διασπαστεί. Θα υπολογισθεί η $h^{x_4}(x_1,x_2)=\begin{Bmatrix} 2, \text{εάν } x_2=0 \,(x_4=1) \\ 0, \text{αλλιώς} \,(x_4=0) \end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_2).

Στο Bucket(x_3), θα σχηματισθούν οι mini-bucket(x_1,x_3) και mini-bucket(x_3,x_5).

- Στο mini-bucket(x_1,x_3):
  - Έχουμε τον $R_{13}(x_1,x_3)$ μόνο του. Θα υπολογισθεί ο $R^{x_3}(x_1)=\{0,1\}$, αλλά αυτός είναι καθολικός και δε θα τοποθετηθεί σε επόμενο κάδο.
  - Έχουμε τη $F_{13}(x_1,x_3)$ μόνη της. Θα υπολογισθεί η $h^{x_3}(x_1)=\begin{Bmatrix} 10, \text{εάν } x_1=0 \,(x_3=0) \\ 2, \text{αλλιώς} \,(x_3=0) \end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_1).
- Στο mini-bucket(x_3,x_5):
  - Έχουμε τον $R_{35}(x_3,x_5)$ μόνο του. Θα υπολογισθεί ο $R^{x_3}(x_5)=\{0,1\}$ αλλά αυτός είναι

καθολικός και δε θα τοποθετηθεί σε επόμενο κάδο.

- Έχουμε τις $F_{235}(x_2,x_3,x_5)$ και $F_3(x_3)$. Προσοχή, η $F_{235}(x_2,x_3,x_5)$ έχει τρεις μεταβλητές αλλά δεν μπορεί να διασπαστεί. Θα υπολογισθεί η $h^{x_3}(x_2,x_5)=\begin{Bmatrix}0,\text{εάν } x_5=1\,(x_3=0)\\8,\text{εάν } x_5=x_2=0\,(x_3=0)\\5,\text{αλλιώς}\,(x_3=1)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_2).

Στο Bucket(x_2), θα σχηματισθούν οι mini-bucket(x_1,x_2) και mini-bucket(x_2,x_5).

- Στο mini-bucket(x_1,x_2):
  - Έχουμε τον $R_{12}(x_1,x_2)$ μόνο του. Θα υπολογισθεί ο $R^{x_2}(x_1)=\{0,1\}$, αλλά αυτός είναι καθολικός και δε θα τοποθετηθεί σε επόμενο κάδο.
  - Έχουμε $h^{x_4}(x_1,x_2)$ και $F_{12}(x_1,x_2)$. Θα υπολογισθεί η $h^{x_2}(x_1)=\begin{Bmatrix}10,\text{εάν } x_1=0\,(x_2=0)\\2,\text{αλλιώς}\,(x_2=0)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_1).
- Στο mini-bucket(x_2,x_5):
  - Έχουμε τον $R_{25}(x_2,x_5)$ μόνο του. Θα υπολογισθεί ο $R^{x_2}(x_5)=\{0,1\}$, αλλά αυτός είναι καθολικός και δε θα τοποθετηθεί σε επόμενο κάδο.
  - Έχουμε τις $h^{x_3}(x_2,x_5)$ και $F_2(x_2)$. Θα υπολογισθεί η $h^{x_2}(x_5)=\begin{Bmatrix}11,\text{εάν } x_5=0\,(x_2=1)\\0,\text{αλλιώς}\,(x_2=0)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_5).

Στο Bucket(x_5), θα σχηματισθεί ο mini-bucket(x_5,x_6).

- Στο mini-bucket(x_5,x_6):
  - Έχουμε τις συναρτήσεις $h^{x_2}(x_5)$, $F_5(x_5)$ και $F_{56}(x_5,x_6)$. Με αυτές θα υπολογισθεί η $h^{x_5}(x_6)=\begin{Bmatrix}17,\text{εάν } x_6=0\,(x_5=0)\\12,\text{αλλιώς}\,(x_5=1)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_6).

Στο Bucket(x_1), θα σχηματισθεί ο mini-bucket(x_1).

- Στο mini-bucket(x_1):
  - Έχουμε τις $h^{x_3}(x_1)$, $h^{x_2}(x_1)$ και $F_1(x_1)$. Θα υπολογισθεί η $h^{x_1}=20\,(x_1=0)$ και θα τοποθετηθεί στον Bucket(x_6).

Στο Bucket(x_6), θα σχηματισθεί ο mini-bucket(x_6).

- Στο mini-bucket(x_6):
  - Έχουμε τις $h^{x_5}(x_6)$, h^{x_1} και $F_6(x_6)$. Θα υπολογισθεί η $h^{x_6}=37\,(x_6=0)$ και θα μας δώσει σαν δεξιό άκρο το 37.

Έχουμε βρει το δεξιό άκρο του διαστήματος, που εκφράζει τη συνάρτηση κόστους. Απομένει να βρούμε την αποτίμηση – λύση του προβλήματος και μέσω αυτής, να υπολογίσουμε το αριστερό άκρο.

Στο Bucket(x_6), πρέπει $x_6 = 0$.
Στο Bucket(x_1), πρέπει $x_1 = 0$.
Στο Bucket(x_5), πρέπει $x_5 = 0$.
Στο Bucket(x_2), πρέπει $x_2 = 1$.
Στο Bucket(x_3), πρέπει $x_3 = 0$.
Στο Bucket(x_4), πρέπει $x_4 = 0$.

Από τις αποτιμήσεις υπολογίζουμε το κάτω όριο:

$F_1(0) + F_2(1) + F_3(0) + F_4(0) + F_5(0) + F_6(0) + F_{12}(0,1) + F_{13}(0,0) + F_{235}(1,0,0) + F_{124}(0,1,0) + F_{56}(0,0) = 0 + 6 + 0 + 0 + 0 + 0 + 0 + 10 + 0 + 0 + 6 = 22$!

Οπότε, η συνάρτηση μας κυμαίνεται στο διάστημα [22,37].

Οι κάδοι στο τέλος:

Κάδος	Συναρτήσεις
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $R_{24}(x_2,x_4)$, $R_{14}(x_1,x_4)$
Bucket(x_3):	$F_3(x_3)$, $F_{235}(x_2,x_3,x_5)$, $F_{13}(x_1,x_3)$, $R_{35}(x_3,x_5)$, $R_{13}(x_1,x_3)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $R_{12}(x_1,x_2)$, $R_{25}(x_2,x_5)$, $h^{x_4}(x_1,x_2)$, $h^{x_3}(x_2,x_5)$
Bucket(x_5):	$F_5(x_5)$, $F_{56}(x_5,x_6)$, $h^{x_2}(x_5)$
Bucket(x_1):	$F_1(x_1)$, $h^{x_3}(x_1)$, $h^{x_2}(x_1)$
Bucket(x_6):	$F_6(x_6)$, $h^{x_5}(x_6)$, h^{x_1}

Πίνακας 9: Η τελική κατάσταση των κάδων μετά το πέρας της MBE-OPT-CONS(i)

5.3 Βιβλιογραφικές αναφορές

Για την πολυπλοκότητα των MBE, μπορείτε να δείτε για περισσότερες πληροφορίες [Lar01]. Για ιστορικούς λόγους, θα ήταν ενδιαφέρον, να δείτε μια άλλη ονομασία των MBE [KLR01].

6 Χρήσεις απαλοιφής με κάδους – μικρο-κάδους

6.1 #P

Εάν κάποιος επισκεφτεί τη ιστοσελίδα του SpringerLink[5] και ψάξει για bucket elimination (ο αγγλικός όρος της απαλοιφής με κάδους), θα πάρει πολλά αποτελέσματα. Από αυτά είναι λίγα που αναφέρονται στους BE σαν θεωρία ή αντικείμενο μελέτης. Τα πιο πολλά αποτελέσματα είναι προβλήματα και πεδία που έχουν εφαρμογή οι BE.

Ένα πολύ ενδιαφέρον πρόβλημα που αντιμετωπίζεται με BE είναι ο υπολογισμός του πλήθους των λύσεων ενός προβλήματος NP. Αυτό το πρόβλημα είναι πάρα πολύ δύσκολο, είναι #P-complete[6]. Οι BE λύνουν τέτοια προβλήματα μετατρέποντας τα σε COP. Για κάθε (ισχυρό) περιορισμό του NP προβλήματος, που αναφέρεται το #P πρόβλημα, αντιστοιχίζεται ένας ασθενής περιορισμός.

Χωρίζουμε τους περιορισμούς σε κάδους και σε κάθε κάδο συνενώνουμε τις σχέσεις – περιορισμούς. Στον περιορισμό R_i που προκύπτει, απαλείφουμε τη μεταβλητή x_p που συμβολίζει ο κάδος Bucket(x_p) και κάθε συνεπή πλειάδα, την αποτιμούμε με το πόσες φορές επαναλαμβάνεται αυτή στο R_i. Μαθηματικά η παραπάνω διαδικασία περιγράφεται ως εξής: Για κάθε περιορισμό του κάδου, αποτιμούμε με ένα τις συνεπείς πλειάδες του. Στον κάδο πολλαπλασιάζουμε τους περιορισμούς και μετά αθροίζουμε ως προς x_p. Εάν συμβολίσουμε με N_i τη συνάρτηση που προκύπτει από τον R_i αποτιμώντας τις συνεπείς πλειάδες του με ένα, υπολογίζουμε το $N^{x_p} = \sum_{x_p} \prod_{N_i \in \text{Bucket}(x_p)} N_i$. Έχουμε πάλι απαλοιφή μεταβλητής αναπαριστώντας τη με κάδους. Μπορούμε να πούμε ότι εάν στον προηγούμενο τύπο το άθροισμα ήταν μεγιστοποίηση και το γινόμενο ήταν άθροισμα, τότε ο αλγόριθμος ELIM-OPT θα έλυνε το πρόβλημά μας. Αυτό και κάνουμε! Δίνουμε έναν αλγόριθμο που είναι παρόμοιος με τον ELIM-OPT.

5 Η SpringerLink είναι μια μηχανή αναζήτησης στο διαδίκτυο για επιστημονικά άρθρα που εκδίδει η ίδια η Springer Verlag.

6 Ουσιαστικά, η κλάση #P ορίζεται από τα προβλήματα υπολογισμού του πλήθους λύσεων προβλημάτων NP και τα #P-complete προβλήματα είναι πιο δύσκολα από τα NP

6.1.1 Αλγόριθμος ELIM-COUNT

Αλγόριθμος ELIM-COUNT
Δεδομένα:
Ένα δίκτυο με περιορισμούς (X,D,C_h), $e = |C_h|$
Μία απαρίθμηση d των μεταβλητών X.
Αρχή Αλγορίθμου:
$N_1,...,N_e$ = οι συναρτήσεις στο Bucket(x_p) που περιγράφουν τους περιορισμούς $R_1,...,R_e$
Για κάθε συνάρτηση N_i, με i από το 1 στο e, **κάνε:**
S_i = η εμβέλεια της N_i
x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει στη S_i
Τοποθέτησε τη N_i στον κάδο Bucket(x_b)
Τέλος Για κάθε
Για κάθε κάδο Bucket(x_p), με p από το n στο 1, **κάνε:**
$N_k,...,N_l$ = οι συναρτήσεις στο Bucket(x_p) που περιγράφουν τους περιορισμούς
$S_k,...,S_l$ = οι εμβέλειες των $N_k,...,N_l$

$$N^{x_p} = \sum_{x_p} \prod_{i=k}^{i=l} N_i$$

x_b = η πιο δεξιά μεταβλητή στην απαρίθμηση d που ανήκει σε $(\bigcup_i S_i) - \{x_p\}$
Τοποθέτησε τη N^{x_p} στο Bucket(x_b)
Τέλος Για κάθε
Τέλος Αλγορίθμου
Έξοδος:
Το πλήθος των λύσεων που υπάρχει στον κάδο Bucket(x_1).

6.1.2 Πολυπλοκότητα ELIM-COUNT

Ο ELIM-COUNT υπολογίζει το πλήθος των λύσεων μέσω VE και είναι BE. Η πολυπλοκότητά του είναι ίδια με αυτή όλων των γνωστών BE.

Θεώρημα: Πολυπλοκότητα ELIM-COUNT

Η χωρική και χρονική πολυπλοκότητα του ELIM-COUNT είναι $O(e \cdot exp(w^*(d))$, όπου e είναι το πλήθος των περιορισμών και d η απαρίθμηση κατά την οποία λύνεται το πρόβλημα.

Απόδειξη:

Για όλους τους BE, η απόδειξη για την πολυπλοκότητά τους είναι η ίδια. Οπότε παραπέμπεστε να δείτε την απόδειξη για την πολυπλοκότητα του MBE-OPT-CONS(i) σε επόμενο κεφάλαιο.

6.1.3 Παράδειγμα υπολογισμού πλήθους λύσεων

Έχοντας παραθέσει τον ELIM-COUNT, θα τον εκτελέσουμε για το στιγμιότυπο του προβλήματος της δημοπρασίας (AP), που είχαμε αναφέρει στην αρχή της εργασίας. Για ευκολία, θα παραθέσουμε εν συντομία το πρόβλημα. Έχουμε:

- ➔ τα προϊόντα $S = \{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8\}$
- ➔ τις δημοπρασίες $B = \{b_1,b_2,b_3,b_4,b_5\}$, όπου
  - $b_1 = \{a_1,a_2,a_3,a_4\}$

- $b_2 = \{a_2,a_3,a_6\}$
- $b_3 = \{a_1,a_4,a_5\}$
- $b_4 = \{a_2,a_8\}$
- $b_5 = \{a_5,a_6\}$

➔ τους περιορισμούς $C = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου
$R_{12} = R_{13} = R_{14} = R_{24} = R_{25} = R_{35} = \{(1,0),(0,1),(0,0)\}$

Θυμίζουμε ότι για κάθε δημοπρασία αντιστοιχούμε μία μεταβλητή. Πριν ξεκινήσουμε την εφαρμογή του ELIM-COUNT, θα μετατρέψουμε τους περιορισμούς σε συναρτήσεις:

- για τον R_{12}, έχουμε τη συνάρτηση $N_{12} = \begin{cases} 1, x_1 = 1 \text{ και } x_2 = 0 \\ 1, x_1 = 0 \text{ και } x_2 = 1 \\ 1, x_1 = 0 \text{ και } x_2 = 0 \\ 0, x_1 = 1 \text{ και } x_2 = 1 \end{cases}$
- για τον R_{13}, έχουμε τη συνάρτηση $N_{13} = \begin{cases} 1, x_1 = 1 \text{ και } x_3 = 0 \\ 1, x_1 = 0 \text{ και } x_3 = 1 \\ 1, x_1 = 0 \text{ και } x_3 = 0 \\ 0, x_1 = 1 \text{ και } x_3 = 1 \end{cases}$
- για τον R_{14}, έχουμε τη συνάρτηση $N_{14} = \begin{cases} 1, x_1 = 1 \text{ και } x_4 = 0 \\ 1, x_1 = 0 \text{ και } x_4 = 1 \\ 1, x_1 = 0 \text{ και } x_4 = 0 \\ 0, x_1 = 1 \text{ και } x_4 = 1 \end{cases}$
- για τον R_{24}, έχουμε τη συνάρτηση $N_{24} = \begin{cases} 1, x_2 = 1 \text{ και } x_4 = 0 \\ 1, x_2 = 0 \text{ και } x_4 = 1 \\ 1, x_2 = 0 \text{ και } x_4 = 0 \\ 0, x_2 = 1 \text{ και } x_4 = 1 \end{cases}$
- για τον R_{25}, έχουμε τη συνάρτηση $N_{25} = \begin{cases} 1, x_2 = 1 \text{ και } x_5 = 0 \\ 1, x_2 = 0 \text{ και } x_5 = 1 \\ 1, x_2 = 0 \text{ και } x_5 = 0 \\ 0, x_2 = 1 \text{ και } x_5 = 1 \end{cases}$
- για τον R_{35}, έχουμε τη συνάρτηση $N_{35} = \begin{cases} 1, x_3 = 1 \text{ και } x_5 = 0 \\ 1, x_3 = 0 \text{ και } x_5 = 1 \\ 1, x_3 = 0 \text{ και } x_5 = 0 \\ 0, x_3 = 1 \text{ και } x_5 = 1 \end{cases}$

Ξεκινάμε τον ELIM-COUNT. Ας επιλέξουμε για απαρίθμηση τη $d = (x_3,x_2,x_1,x_5,x_4)$. Παρακάτω, φαίνονται οι κάδοι αρχικά.

Κάδος	Συναρτήσεις
Bucket(x_4):	$N_{14}(x_1,x_4)$
Bucket(x_5):	$N_{25}(x_2,x_5)$, $N_{35}(x_3,x_5)$
Bucket(x_1):	$N_{12}(x_1,x_2)$, $N_{13}(x_2,x_4)$
Bucket(x_2):	$N_{23}(x_2,x_3)$
Bucket(x_3):	

Πίνακας 10: Οι κάδοι μετά την αρχικοποίηση του ELIM-COUNT

Στο Bucket(x_4), απλά θα κάνουμε απαλοιφή ως προς x_4,

$N^{x_4}(x_1)=\sum_{x_4} N_{14}(x_1,x_4)=\begin{Bmatrix}2, x_1=0\\ 1, x_1=1\end{Bmatrix}$ και τοποθετούμε τη $N^{x_4}(x_1)$ στο Bucket(x_1).

Στο Bucket(x_5) θα πολλαπλασιάσουμε τις $N_{25}(x_2,x_5)$ και $N_{35}(x_3,x_5)$ και θα κάνουμε απαλοιφή ως προς x_5, $N^{x_5}(x_2,x_3)=\sum_{x_5}\prod_{N_i\in \text{Bucket}(x_5)} N_i=\begin{Bmatrix}2, x_2=0 \text{ και } x_3=0\\ 1, x_2=0 \text{ και } x_3=1\\ 1, x_2=1 \text{ και } x_3=0\\ 1, x_2=1 \text{ και } x_3=1\end{Bmatrix}$.Τοποθετούμε τη $N^{x_5}(x_2,x_3)$ στο Bucket(x_2).

Στο Bucket(x_1), θα πολλαπλασιάσουμε τις $N_{12}(x_1,x_2)$, $N_{13}(x_1,x_3)$ και $N^{x_4}(x_1)$. Θα κάνουμε απαλοιφή ως προς x_1, $N^{x_1}(x_2,x_3)=\sum_{x_1}\prod_{N_i\in \text{Bucket}(x_1)} N_i=\begin{Bmatrix}3, x_2=0 \text{ και } x_3=0\\ 2, x_2=0 \text{ και } x_3=1\\ 2, x_2=1 \text{ και } x_3=0\\ 2, x_2=1 \text{ και } x_3=1\end{Bmatrix}$ και τοποθετούμε τη $N^{x_1}(x_2,x_3)$ στο Bucket(x_2).

Στο Bucket(x_2), θα πολλαπλασιάσουμε τις $N_{23}(x_2,x_3)$, $N^{x_5}(x_2,x_3)$ και $N^{x_1}(x_2,x_3)$. Θα κάνουμε απαλοιφή ως προς x_2, $N^{x_2}(x_3)=\sum_{x_2}\prod_{N_i\in \text{Bucket}(x_2)} N_i=\begin{Bmatrix}8, x_3=0\\ 2, x_3=1\end{Bmatrix}$ και τοποθετούμε τη $N^{x_2}(x_3)$ στο Bucket(x_3).

Στο Bucket(x_3), θα κάνουμε απαλοιφή ως προς x_3, $N^{x_3}=\sum_{x_3} N^{x_2}(x_3)=10$. Έχουμε 10 λύσεις! Οι κάδοι μετά το τέλος του αλγορίθμου:

Κάδος	Συναρτήσεις
Bucket(x_4):	$N_{14}(x_1,x_4)$
Bucket(x_5):	$N_{25}(x_2,x_5)$, $N_{35}(x_3,x_5)$
Bucket(x_1):	$N_{12}(x_1,x_2)$, $N_{13}(x_2,x_4)$, $N^{x_4}(x_1)$
Bucket(x_2):	$N_{23}(x_2,x_3)$, $N^{x_5}(x_2,x_3)$, $N^{x_1}(x_2,x_3)$
Bucket(x_3):	$N^{x_2}(x_3)$

Πίνακας 11: Οι κάδοι μετά το πέρας του ELIM-COUNT

Ας κάνουμε κάτι ενδιαφέρον, ας επαληθεύσουμε ότι οι λύσεις του προβλήματος είναι δέκα. Θα βρούμε όλες τις πιθανές αποτιμήσεις, θα δούμε ποιες είναι συνεπείς και θα τις προσθέσουμε. Σημειωτέον, ότι αυτή η διαδικασία δεν είναι εφικτή για πολλές μεταβλητές, ο αριθμός των δυνατών αποτιμήσεων είναι εκθετικός ως προς τον αριθμό των μεταβλητών.

x_1	x_2	x_3	x_4	x_5	$\vec{a}(x1,x2,x3,x4,x5)$
0	0	0	0	0	***συνεπής***
0	0	0	0	1	***συνεπής***
0	0	0	1	0	***συνεπής***
0	0	0	1	1	***συνεπής***
0	0	1	0	0	***συνεπής***
0	0	1	0	1	ασυνεπής
0	0	1	1	0	***συνεπής***
0	0	1	1	1	ασυνεπής
0	1	0	0	0	***συνεπής***
0	1	0	0	1	ασυνεπής
0	1	0	1	0	***συνεπής***
0	1	0	1	1	ασυνεπής
0	1	1	0	0	ασυνεπής
0	1	1	0	1	ασυνεπής
0	1	1	1	0	ασυνεπής
0	1	1	1	1	ασυνεπής
1	0	0	0	0	***συνεπής***
1	0	0	0	1	***συνεπής***
1	0	0	1	0	ασυνεπής
1	0	0	1	1	ασυνεπής
1	0	1	0	0	ασυνεπής
1	0	1	0	1	ασυνεπής
1	0	1	1	0	ασυνεπής
1	0	1	1	1	ασυνεπής
1	1	0	0	0	ασυνεπής
1	1	0	0	1	ασυνεπής
1	1	0	1	0	ασυνεπής
1	1	0	1	1	ασυνεπής
1	1	1	0	0	ασυνεπής
1	1	1	0	1	ασυνεπής
1	1	1	1	0	ασυνεπής
1	1	1	1	1	ασυνεπής

Πίνακας 12: Όλες οι πιθανές αποτιμήσεις για το AP πρόβλημά μας

Δικαιωθήκαμε!

6.2 Αλγόριθμοι πάσης στιγμής

Η απαλοιφή με μικρο-κάδους (MBE) έχει βρει μεγάλη απήχηση στο πεδίο του CP. Πολλοί αλγόριθμοι, που προϋπήρχαν στο χώρο, έχουν ενσωματώσει τη MBE βελτιώνοντας την απόδοσή τους και άλλοι καινούργιοι έχουν ανακαλυφθεί, όντας εμπνευσμένοι από τη MBE. Θα δούμε μία τέτοια έμπνευση και είναι πολύ πιθανόν, ότι πριν παραθέσω τον καινούργιο αλγόριθμο, θα τον έχετε ήδη σκεφτεί πρώτα εσείς! Συνεχίζουμε με τον ορισμό μιας οικογένειας αλγορίθμων.

Τις τελευταίες δύο με τρεις δεκαετίες έχουν οριστεί οι αλγόριθμοι πάσης στιγμής[7]. Αυτοί οι αλγόριθμοι είναι κατασκευασμένοι με τέτοιο τρόπο, που όταν τους διακόψουμε προσωρινά, αυτοί μας βγάζουν στην έξοδο το καλύτερο αποτέλεσμα που έχουν υπολογίσει έως εκείνη τη στιγμή. Αφού το κάνουν αυτό, μπορούμε να τους αφήσουμε να συνεχίσουν να υπολογίζουν. Εάν τους διακόψουμε ξανά, οφείλουν να μας επιστρέψουν ένα αποτέλεσμα και αυτό να είναι καλύτερο από το προηγούμενο!

Αυτοί οι αλγόριθμοι εκτελούνται επ' άπειρον και κάθε φορά που τους το ζητάμε, μας επιστρέφουν ένα αποτέλεσμα που είναι το ίδιο καλό ή καλύτερο από τα προηγούμενα. Αυτή είναι η βασική αρχή των αλγόριθμων πάσης στιγμής.

Υπάρχει ολόκληρη θεωρία πίσω από αυτή την οικογένεια αλγορίθμων, αλλά δε μας ενδιαφέρει κάτι από αυτά στη μελέτη μας. Το μόνο που αξίζει να αναφέρουμε και θα το επιδιώξουμε, είναι ότι πολλές φορές αρκούμαστε σε μια λύση που είναι μέσα στα όρια ενός προσεγγιστικού σφάλματος ε. Εάν πετύχουμε την προσέγγιση αυτή, τερματίζουμε τον αλγόριθμο. Υπάρχουν πολλά χαρακτηριστικά των αλγορίθμων πάσης στιγμής, αλλά εμείς θα επικεντρωθούμε στα επόμενα:

Ορισμός: Αλγόριθμος πάσης στιγμής

Αλγόριθμος πάσης στιγμής (anytime algorithm) καλείται ο αλγόριθμος που έχει τις εξής απλές ιδιότητες:

- ✔ *Μπορούμε ανά πάσα στιγμή να τον παύσουμε και οφείλει, όταν γίνεται αυτό, να μας δώσει ένα αποτέλεσμα.*
- ✔ *Μετά την παύση τους, οι αλγόριθμοι αυτοί πρέπει να συνεχίζουν τον υπολογισμό τους.*
- ✔ *Κάθε φορά που θα διακόπτουμε τον αλγόριθμο, πρέπει να μας δίνει ένα αποτέλεσμα που είναι το ίδιο καλό ή καλύτερο από το προηγούμενο.*

6.2.1 Αλγόριθμος anytime-mbe

Πιστεύω πως αυτή τη στιγμή έχετε σκεφτεί ότι οι MBE μπορούν να χρησιμοποιηθούν σαν αλγόριθμοι πάσης στιγμής. Η διαδικασία είναι απλή, τρέχουμε κάθε φορά το MBE για διαφορετικό i. Αρχίζουμε με i = 1, μετά i = 2, ... και συνεχίζουμε έως ότου θέλουμε. Φανταστικό και απλό. Πρέπει να βεβαιωθούμε ότι μια τέτοια εκτέλεση μπορεί να πληρεί τα κριτήρια των αλγορίθμων πάσης στιγμής, όπως αυτά δηλώνονται στον ορισμό, αλλά πρώτα ας παραθέσουμε ένα αλγόριθμο που να το επιτυγχάνει.

7 Στη βιβλιογραφία, τους αλγόριθμους πάσης στιγμής, μπορούμε να τους βρούμε και σαν «αλγόριθμους οποιουδήποτε χρόνου».

Αλγόριθμος anytime-mbe(ε)
Δεδομένα:
Μία αρχική τιμή i_0 για το i.
Το βήμα(step) i_{step} με το οποίο θα αυξάνεται το i.
Το σφάλμα προσέγγισης (approximation error) ε.
Αρχή Αλγορίθμου:
$i = i_0$
U = η μικρότερη τιμή που μπορεί να πάρει θεωρητικά η συνάρτηση κόστους
L = η μικρότερη τιμή που μπορεί να πάρει θεωρητικά η συνάρτηση κόστους
Επανέλαβε
εκτέλεσε τον MBE-OPT-CONS(i)
U_{temp} = το δεξιό άκρο του διαστήματος που υπολογίζει ο MBE-OPT-CONS(i) σαν συνάρτηση κόστους
Εάν $U_{temp} > U$, **τότε**
$U = U_{temp}$
Τέλος εάν
L_{temp} = το αριστερό άκρο του διαστήματος που υπολογίζει ο MBE-OPT-CONS(i) σαν συνάρτηση κόστους
Εάν $L_{temp} > L$, **τότε**
$L = L_{temp}$
Τέλος εάν
Εάν $1 \leqslant \frac{U}{L} \leqslant 1+\varepsilon$, **τότε**
ο αλγόριθμος τερματίζει
Αλλιώς
$i = i + i_{step}$
Τέλος εάν
Μέχρι να τελειώσουν οι υπολογιστικοί πόροι της μηχανής.
Τέλος Αλγορίθμου
Έξοδος:
Ένα κάτω φράγμα L και ένα άνω φράγμα U για τη συνάρτηση κόστους

Ο παραπάνω αλγόριθμος θα μπορούσε να επιστρέφει και τη «βέλτιστη» αποτίμηση, την αποτίμηση που επιφέρει το βέλτιστο κόστος, αλλά δεν το κάνει για λόγους απλότητας. Σκοπός είναι να φανεί απλά η μεθοδολογία του αλγορίθμου, το πως οι MBE γίνονται αλγόριθμοι πάσης στιγμής.

Απομένει να δείξουμε ότι ο anytime-mbe(ε) είναι αλγόριθμος πάσης στιγμής. Θα διερευνήσουμε τις τρεις προϋποθέσεις – ιδιότητες του ορισμού του αλγορίθμου πάσης στιγμής.

- ✔ Ο αλγόριθμος μπορεί να διακοπεί ανά πάσα στιγμή και να δώσει ένα αποτέλεσμα, το L και το U. Εννοείται ότι δεν είμαστε κακόβουλοι και δε σταματάμε τον αλγόριθμο με το που περάσει ένα δευτερόλεπτο.
- ✔ Ο αλγόριθμος θα συνεχίσει από εκεί που διακόπηκε. Στην πραγματικότητα δε θα σταματήσει τελείως, θα δώσει τα μέχρι εκείνη τη στιγμή αποτελέσματα και θα συνεχίζει παράλληλα τους υπολογισμούς.
- ✔ Πρέπει να δείξουμε ότι κάθε φορά ο anytime-mbe(ε) θα επιστρέφει καλύτερο αποτέλεσμα. Για να μην ισχύει η προηγούμενη πρόταση, πρέπει ο αλγόριθμος για κάποια τιμή του i, i_{evil}, να μην προσεγγίζει καλύτερα το κόστος από τα L_{evil} και U_{evil}. Αυτό δεν ισχύει καθώς όταν το i φτάσει την τιμή $w^*(d)$, τότε ο MBE-OPT-CONS(i) θα επιστρέψει το βέλτιστο κόστος. Το μόνο πρόβλημα που μπορεί να προκύψει, είναι να τελειώσουν οι υπολογιστικοί πόροι, αλλά

για αυτό δε φταίει ο αλγόριθμος.

6.2.2 Κακός αλγόριθμος πάσης στιγμής

Όπως έχουμε πει, η θεωρία για αλγόριθμους πάσης στιγμής είναι τεράστια. Με τον καιρό, έχουν προστεθεί και άλλες ιδιότητες που θα ήταν καλό να έχουν αυτοί οι αλγόριθμοι. Όσον αφορά το σχόλιο ότι μπορεί να τελειώσουν οι πόροι για το anytime-mbe(ε), θα τελειώσουμε την ανάλυσή μας για αλγόριθμους πάσης στιγμής με το εξής:

Για παράδειγμα, θεωρητικά, ο επόμενος αλγόριθμος είναι αλγόριθμος πάσης στιγμής, αλλά δεν έχει κανένα ενδιαφέρον και δεν θα πρέπει να αναφέρεται ως έτσι:

Αλγόριθμος anytime-fool
Αρχή Αλγορίθμου:
M = η προσεγγιστική τιμή της συνάρτηση κόστους
Επανέλαβε
Για κάθε δυνατή αποτίμηση $\vec{a}$, **κάνε**
Εάν η αποτίμηση αυτή είναι συνεπής, **τότε**
M_{temp} = το κόστος που αποφέρει η $\vec{a}$
Εάν $M_{temp} > M$, **τότε**
$M = M_{temp}$
Τέλος εάν
Τέλος εάν
Τέλος Για κάθε
Μέχρι να τελειώσουν οι υπολογιστικοί πόροι της μηχανής.
Τέλος Αλγορίθμου
Έξοδος:
Μια προσεγγιστική τιμή για τη συνάρτηση κόστους

Εύκολα δείχνεται ότι ο παραπάνω αλγόριθμος πληρεί τις τρεις προϋποθέσεις που απαιτήσαμε. Υπάρχουν άλλες πολλές, στο χώρο των αλγορίθμων πάσης στιγμής, που δεν τις πληρεί. Μία προϋπόθεση που δεν πληρεί, και την έχουμε δει στο anytime-mbe(ε), είναι ότι δεν υπάρχει πουθενά κάποιο όριο σφάλματος που να καθοδηγεί τον αλγόριθμο.

6.3 Αναζήτηση με επέκταση και οριοθέτηση

Ίσως η μεγαλύτερη συνεισφορά των MBE, είναι η χρήση τους σαν ευριστική συνάρτηση (heuristic function). Οι MBE είναι αλγόριθμοι συμπερασμού. Μπορούμε να χρησιμοποιήσουμε MBE σε συνδυασμό με αλγορίθμους αναζήτησης. Πριν δούμε πως γίνεται αυτό, ας δούμε για αλγορίθμους αναζήτησης γενικότερα.

Υπάρχουν αρκετοί αλγόριθμοι αναζήτησης, αλλά για COP ο πιο γνωστός είναι η αναζήτηση με επέκταση και οριοθέτηση (branch and bound search). Στη βιβλιογραφία, ο αλγόριθμος είναι δημοφιλής με τα αρχικά BnB[8] από το αγγλικό όρο branch and bound. Αυτά τα αρχικά, για χάρη συντομίας, θα χρησιμοποιούμε όταν αναφερόμαστε σε αναζήτηση με επέκταση και οριοθέτηση.

Θα μπορούσαμε να είμαστε πιο αναλυτικοί περιγράφοντας τη BnB, γράφοντας μία διπλωματική εργασία μόνο για αυτήν. Σε αυτή την εργασία θα αναφερθούμε γενικά στη BnB, παραλείποντας αρκετά σημαντικά πράγματα, αλλά αναφέροντας οπωσδήποτε όλα όσα μας ενδιαφέρουν.

Μπορούμε να σκεφτούμε τη BnB σαν μια πρώτα κατά βάθος αναζήτηση (DFS) σε ένα

8 Το γράμμα n στο ακρωνύμιο BnB συμβολίζει τη λέξη and. Είναι σύνηθες φαινόμενο το and να το βρίσκουμε σαν n σε πολλά πρόχειρα κείμενα που θέλουμε να αποφύγουμε τη γραφή πολλών γραμμάτων. Κυρίως, αυτό συμβαίνει σε συνομιλίες μέσω γραπτού κειμένου στο διαδίκτυο, τα γνωστά μας «chat».

δέντρο αναζήτησης. Τι είναι δέντρο αναζήτησης;

Ορισμός: Δέντρο αναζήτησης

> *Δέντρο αναζήτησης (search tree) είναι ένα δέντρο που κατασκευάζουμε για ένα πρόβλημα με περιορισμούς (CP). Ορίζουμε μια απαρίθμηση d για τις n μεταβλητές του προβλήματος. Το δέντρο αναζήτησης έχει σαν κόμβους τιμές για τις μεταβλητές. Στο πρώτο επίπεδο του δέντρου, υπάρχουν τιμές για την πρώτη μεταβλητή στη d, στο επόμενο για τη δεύτερη, και συνεχίζουμε μέχρι να φτάσουμε στα φύλλα του δέντρου όπου υπάρχουν τιμές για τη n-οστή μεταβλητή.*

Η BnB είναι μια DFS. Κάθε φορά που επισκέπτεται ένα φύλλο του δέντρου αναζήτησης, υπολογίζει το κόστος της διαδρομής που έχει διανύσει από τη ρίζα μέχρι εκείνο το φύλλο. Αυτό το συγκρίνει με το βέλτιστο ως εκείνη τη στιγμή κόστος, και ανάλογα ανανεώνει την τιμή του έως τότε βέλτιστου κόστους. Όταν θα έχει επισκεφτεί όλα τα φύλλα του δέντρου, θα έχει βρει το ολικό βέλτιστο κόστος.

Είπαμε η BnB είναι μια DFS. Αλλά το να τρέξουμε μια DFS σε ένα δέντρο αναζήτησης κοστίζει $O(k^n)$, όπου k είναι η πληθικότητα του μεγαλύτερο πεδίου τιμών των μεταβλητών. Προφανώς η BnB δεν είναι ένα απλός DFS αλγόριθμος, γιατί τότε θα ήταν πιο αργός και από την καθυστέρηση.

Για να αποφύγουμε να επισκεφθούμε όλους τους κόμβους του δέντρου, την πρώτη φορά που θα επισκεφθούμε ένα φύλλο του δέντρου, κρατάμε τη τιμή του κόστους σε μία μεταβλητή L. Έστω ότι η BnB είναι σε έναν εσωτερικό κόμβο x_p. Εάν είχαμε ένα μαντείο (oracle) που να μας μαρτυρούσε πόσο είναι το κόστος $f^*(\vec{a}_p)$ για το μονοπάτι από τη ρίζα στα φύλλα μέσω του x_p, για κάθε τιμή του x_p, θα αποκλείαμε να επισκεφτούμε όλα τα κλαδιά του δέντρου – τιμές του x_p – που μας δίνουν $f^*(\vec{a}_p)$ που είναι μικρότερο του L, $\vec{a}_p$ είναι η αποτίμηση της BnB μέχρι τον κόμβο x_p. Κάθε φορά που επισκεπτόμαστε ένα φύλλο ανανεώνουμε τη τιμή του L, εφόσον το κόστος σε αυτό το φύλλο είναι μεγαλύτερο του L.

Δυστυχώς, μαντεία δεν υπάρχουν, ή δεν έχουμε καταφέρει να κατασκευάσουμε ακόμα. Αυτό που μπορούμε να κάνουμε, είναι να εκτιμήσουμε αυτό το κόστος και να αποφασίσουμε πιο κλαδί του δέντρου, δηλαδή ποια τιμή της μεταβλητής, θα διαλέξουμε. Το $f^*(\vec{a}_p)$ τώρα δεν υπολογίζεται από μαντείο και ονομάζεται συνάρτηση εκτίμησης οριοθέτησης (bounding evaluation function) γιατί εκτιμά το κόστος και δεν το υπολογίζει ακριβώς. Προσοχή, μπορούμε να υπερεκτιμήσουμε όσο θέλουμε αυτό το κόστος, αλλά δεν επιτρέπεται με τίποτα να το υποεκτιμήσουμε γιατί τότε μπορεί να αποκλείσουμε την αναζήτηση σε κλαδιά χωρίς αυτό να είναι σωστό.

Όλα τα λεφτά είναι η κατασκευή της $f^*(\vec{a}_p)$. Όσο καλύτερη είναι η $f^*(\vec{a}_p)$, τόσο περισσότερα κλαδιά θα αποκλείουμε. Θα έχετε ήδη φανταστεί ότι το ρόλο της $f^*(\vec{a}_p)$ θα τον παίξει ο MBE.

6.3.1 Ο αλγόριθμος απαλοιφής μικρο-κάδων σαν ευριστική συνάρτηση

Η $f^*(\vec{a}_p)$ στον κόμβο x_p υπολογίζει το κόστος από τη ρίζα του δέντρου, ακολουθώντας τη διαδρομή της αποτίμησης $\vec{a}_p$ και μαντεύοντας το μονοπάτι που αποδίδει τη βέλτιστη συνάρτηση κόστους. Θαύματα δεν κάνουμε, μάντεις δεν είμαστε και θα πρέπει να προσεγγίσουμε το κόστος για το μονοπάτι μετά τον κόμβο x_p.

Να κάνουμε λίγες πράξεις. Με buckets(i..j) αναπαριστούμε όλους τους κάδους από τον Bucket(x_i) μέχρι τον κάδο Bucket(x_j).

$$f^*(\vec{a}_p) = \max_{a_{p+1},\dots,a_n} \sum_k F_k(\vec{a}_p, a_{p+1}, \dots, a_n) =$$

$$f^*(\vec{a}_p)=\max_{a_{p+1},\dots,a_n}\left\{\sum_{F_i\in buckets(1..p)}F_i(\vec{a}_p)+\sum_{F_i\in buckets(p+1..n)}F_i(\vec{a}_p,a_{p+1},\dots,a_n)\right\}=$$

$$f^*(\vec{a}_p)=\sum_{F_i\in buckets(1..p)}F_i(\vec{a}_p)+\max_{a_{p+1},\dots,a_n}\sum_{F_i\in buckets(p+1..n)}F_i(\vec{a}_p,a_{p+1},\dots,a_n)=g(\vec{a}_p)+h^*(\vec{a}_p)\ ,$$

όπου

$$g(\vec{a}_p)=\sum_{F_i\in buckets(1..p)}F_i(\vec{a}_p)$$

και

$$h^*(\vec{a}_p)=\max_{a_{p+1},\dots,a_n}\sum_{F_i\in buckets(p+1..n)}F_i(\vec{a}_p,a_{p+1},\dots,a_n)\ .$$

Οπότε αρκεί να βρούμε μία «καλή» $h^*(\vec{a}_p)$. Αλλά, όπως όλοι θα έχετε μυριστεί, την $h^*(\vec{a}_p)$, την έχουμε υπολογίσει υπερεκτιμώντας τη με απαλοιφή με κάδους!

Ορισμός: Ευριστική συνάρτηση «μικρο-κάδος»

> *Έστω ότι έχουμε μία απαρίθμηση d των μεταβλητών. Η συνάρτηση ευριστικής οριοθέτησης «μικρο-κάδος»* $h(\vec{a}_p)$ *ισούται με το άθροισμα όλων των καινούργιων συναρτήσεων που δημιουργήθηκαν στους κάδους buckets(p+1..n) και έχουν τοποθετηθεί στους κάδους buckets(1..p) κατά τον MBE. Έχουμε:*

$$h(\vec{a}_p)=\sum_{i=1}^{p}\sum_{h^k{}_j\in Bucket(x_i)}h^k{}_j\,,\ \mu\varepsilon\ k>p$$

Μένει να δείξουμε ότι η $h(\vec{a}_p)$ υπερεκτιμά τη $h^*(\vec{a}_p)$ και ότι κάθε φορά που πλησιάζουμε στα φύλλα, δηλαδή μεγαλώνει το p, αυτή ελαττώνεται ώστε να πλησιάζουμε το βέλτιστο κόστος και να μην απομακρυνόμαστε από αυτό.

Υπάρχει αυστηρή μαθηματική απόδειξη για την παραπάνω παράγραφο, αλλά ξεφεύγει από τα πλαίσια αυτής της εργασίας. Θα το δείξουμε διαισθητικά. Πρέπει να δείξουμε ότι η $h(\vec{a}_p)$ υπερεκτιμά τη $h^*(\vec{a}_p)$. Αυτό είναι αλήθεια γιατί από τον MBE οι συναρτήσεις, που παράγονται από τους κάδους, είναι μεγαλύτερες ή ίσες με το βέλτιστο κόστος που θα παραγόταν από αυτούς τους κάδους. Το ότι κάθε φορά η $h(\vec{a}_p)$ ελαττώνεται, ισχύει γιατί όσο αυξάνει, τόσο μειώνονται οι κάδοι που παίρνουμε υπόψιν μας (k>p).

6.3.2 Παράδειγμα απαλοιφής με μικρο-κάδους σαν ευριστική συνάρτηση.

Θα τρέξουμε τη BnB με ευριστική συνάρτηση την απαλοιφή με μικρο-κάδους, για το γνωστό μας παράδειγμα. Για συντομία, θα παραθέσουμε γρήγορα το παράδειγμα εδώ.

- ➔ $X = \{x_1,x_2,x_3,x_4,x_5,x_6\}$.
- ➔ $D = \{D_1,D_2,D_3,D_4,D_5,D_6\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = \{0,1\}$.
- ➔ Ασθενείς περιορισμοί C_s:
  - $F_1(x_1)=\begin{cases}8, \text{εάν } x_1=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_2(x_2)=\begin{cases}6, \text{εάν } x_2=1\\ 0, \text{αλλιώς}\end{cases}$
  - $F_3(x_3)=\begin{cases}5, \text{εάν } x_3=1\\ 0, \text{αλλιώς}\end{cases}$

- $F_4(x_4)=\begin{Bmatrix}2,\text{εάν } x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_5(x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_6(x_6)=\begin{Bmatrix}4,\text{εάν } x_6=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{12}(x_1,x_2)=\begin{Bmatrix}10,\text{εάν } x_1=x_2=1\\8,\text{εάν } x_1=x_2=0\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{13}(x_1,x_3)=\begin{Bmatrix}8,\text{εάν } x_1=x_3=1\\2,\text{εάν } x_1=1 \text{ και } x_3=0\\10,\text{εάν } x_1=0 \text{ και } x_3=0\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix}8,\text{εάν } x_2=x_3=x_5\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix}8,\text{εάν } x_1=x_2=x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
- $F_{56}(x_5,x_6)=\begin{Bmatrix}10,\text{εάν } x_5=x_6=1\\6,\text{εάν } x_5=x_6=0\\0,\text{αλλιώς}\end{Bmatrix}$

➔ Ισχυροί περιορισμοί, C_h = $\{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}=R_{13}=R_{14}=R_{24}=R_{25}=R_{35}=\{(1,0),(0,1),(0,0)\}$.

➔ Τα αποτελέσματα του MBE-OPT-CONS(2) κατά τη d = $(x_6,x_1,x_5,x_2,x_3,x_4)$,

- Στο Bucket(x_4):
  -
- Στο Bucket(x_3):
  -
- Στο Bucket(x_2):
  $h^{x_4}(x_1,x_2)=\begin{Bmatrix}2,\text{εάν } x_2=0\,(x_4=1)\\0,\text{αλλιώς}\,(x_4=0)\end{Bmatrix}$
  $h^{x_3}(x_2,x_5)=\begin{Bmatrix}0,\text{εάν } x_5=1\,(x_3=0)\\8,\text{εάν } x_5=x_2=0\,(x_3=0)\\5,\text{αλλιώς}\,(x_3=1)\end{Bmatrix}$
- Στο Bucket(x_5):
  $h^{x_2}(x_5)=\begin{Bmatrix}11,\text{εάν } x_5=0\,(x_2=1)\\0,\text{αλλιώς}\,(x_2=0)\end{Bmatrix}$
- Στο Bucket(x_1):
  $h^{x_3}(x_1)=\begin{Bmatrix}10,\text{εάν } x_1=0\,(x_3=0)\\2,\text{αλλιώς}\,(x_3=0)\end{Bmatrix}$
  $h^{x_2}(x_1)=\begin{Bmatrix}10,\text{εάν } x_1=0\,(x_2=0)\\2,\text{αλλιώς}\,(x_2=0)\end{Bmatrix}$
- Στο Bucket(x_6):
  $h^{x_5}(x_6)=\begin{Bmatrix}17,\text{εάν } x_6=0\,(x_5=0)\\12,\text{αλλιώς}\,(x_5=1)\end{Bmatrix}$

$h^{x_1} = 20 \, (x_1 = 0)$

Ξεκινάμε τη BnB σύμφωνα με την απαρίθμηση d. Θυμίζουμε $f(\vec{a_p}) = g(a_p) + h(\vec{a_p})$.

- x_6
  - $x_6 = 0$
    - $g = F_6(0) = 0$
    - $h = h^{x_5}(x_6) + h^{x_1} = 17 + 20 = 37$
    - $f = g + h = 0 + 37 = 37$
  - $x_6 = 1$
    - $g = F_6(1) = 4$
    - $h = h^{x_5}(x_6) + h^{x_1} = 12 + 20 = 32$
    - $f = g + h = 4 + 32 = 36$

  Άρα επιλέγουμε $x_6 = 0$.
- x_1
  - $x_1 = 0$
    - $g = 0 + F_1(0) = 0 + 0 = 0$
    - $h = h^{x_5}(0) + h^{x_3}(x_1) + h^{x_2}(x_1) = 17 + 10 + 10 = 37$
    - $f = g + h = 0 + 37 = 37$
  - $x_1 = 1$
    - $g = 0 + F_1(1) = 0 + 8 = 8$
    - $h = h^{x_5}(0) + h^{x_3}(x_1) + h^{x_2}(x_1) = 17 + 2 + 2 = 21$
    - $f = g + h = 8 + 21 = 29$

  Άρα επιλέγουμε $x_1 = 0$.
- x_5
  - $x_5 = 0$
    - $g = 0 + F_5(0) + F_{56}(0,0) = 0 + 0 + 6 = 6$
    - $h = h^{x_3}(0) + h^{x_2}(0) + h^{x_2}(x_5) = 10 + 10 + 11 = 31$
    - $f = g + h = 6 + 31 = 37$
  - $x_5 = 1$
    - $g = 0 + F_5(1) + F_{56}(1,0) = 0 + 2 + 0 = 2$
    - $h = h^{x_3}(0) + h^{x_2}(0) + h^{x_2}(x_5) = 10 + 10 + 0 = 20$
    - $f = g + h = 2 + 20 = 22$

  Άρα επιλέγουμε $x_5 = 0$.
- x_2
  - $x_2 = 0$
    - $g = 6 + F_{12}(0,0) + F_2(0) = 6 + 8 + 0 = 14$
    - $h = h^{x_3}(0) + h^{x_4}(0, x_2) + h^{x_3}(x_2, 0) = 10 + 2 + 8 = 20$
    - $f = g + h = 14 + 20 = 34$
  - $x_2 = 1$
    - $g = 6 + F_{12}(0,1) + F_2(1) = 6 + 0 + 6 = 12$
    - $h = h^{x_3}(0) + h^{x_4}(0, x_2) + h^{x_3}(x_2, 0) = 10 + 0 + 5 = 15$
    - $f = g + h = 12 + 15 = 27$

  Άρα επιλέγουμε $x_2 = 0$.
- x_3
  - $x_3 = 0$
    - $g = 14 + F_{13}(0,0) + F_{235}(0,0,0) + F_3(0) = 14 + 10 + 8 + 0 = 32$
    - $h = h^{x_4}(0,0) = 2$
    - $f = g + h = 32 + 2 = 34$
  - $x_3 = 1$

- $g = 14 + F_{13}(0,1) + F_{235}(0,1,0) + F_3(1) = 14 + 0 + 0 + 5 = 19$
- $h = h^{x_4}(0,0) = 2$
- $f = g + h = 15 + 2 = 17$

Άρα επιλέγουμε $x_3 = 0$.

- x_4
  - $x_4 = 0$
    - $g = 32 + F_{124}(0,0,0) + F_4(0) = 32 + 0 + 0 = 32$
    - $h = 0$
    - $f = g + h = 32 + 0 = 32$
  - $x_4 = 1$
    - $g = 32 + F_{124}(0,0,1) + F_4(1) = 32 + 0 + 2 = 34$
    - $h = 0$
    - $f = g + h = 34 + 0 = 34$

  Άρα επιλέγουμε $x_4 = 1$.
  Βρήκαμε και το πρώτο κάτω όριο L = 34. Η BnB οπισθοδρομεί και πηγαίνει να επιλέξει $x_3 = 1$.
- x_4
  - $x_4 = 0$
    - $g = 19 + F_{124}(0,0,0) + F_4(0) = 19 + 0 + 0 = 19$
    - $h = 0$
    - $f = g + h = 19 + 0 = 19$
  - $x_4 = 1$
    - $g = 19 + F_{124}(0,0,1) + F_4(1) = 19 + 0 + 2 = 21$
    - $h = 0$
    - $f = g + h = 21 + 0 = 21$

  Έχουμε $f = 21 < L$. Οπισθοδρομούμε και άλλο, $x_2 = 1$.
- x_3
  - $x_3 = 0$
    - $g = 12 + F_{13}(0,0) + F_{235}(0,0,0) + F_3(0) = 12 + 10 + 8 + 0 = 30$
    - $h = h^{x_4}(0,1) = 0$
    - $f = g + h = 30 + 0 = 30$
  - $x_3 = 1$
    - $g = 12 + F_{13}(0,1) + F_{235}(0,1,0) + F_3(1) = 12 + 0 + 0 + 5 = 17$
    - $h = h^{x_4}(0,1) = 0$
    - $f = g + h = 17 + 0 = 17$

  Έχουμε $f = 30 < L$. Οπισθοδρομούμε και άλλο, $x_5 = 1$.
- x_2
  - $x_2 = 0$
    - $g = 2 + F_{12}(0,0) + F_2(0) = 2 + 8 + 0 = 10$
    - $h = h^{x_3}(0) + h^{x_4}(0,x_2) + h^{x_3}(x_2,1) = 10 + 2 + 0 = 12$
    - $f = g + h = 10 + 12 = 22$
  - $x_2 = 1$
    - $g = 2 + F_{12}(0,1) + F_2(1) = 2 + 0 + 6 = 8$
    - $h = h^{x_3}(0) + h^{x_4}(0,x_2) + h^{x_3}(x_2,1) = 10 + 0 + 0 = 10$
    - $f = g + h = 8 + 10 = 18$

  Έχουμε $f = 22 < L$. Οπισθοδρομούμε και άλλο, $x_1 = 1$.
- x_5
  - $x_5 = 0$
    - $g = 8 + F_5(0) + F_{56}(0,0) = 8 + 0 + 6 = 14$

        - $h = h^{x_3}(1) + h^{x_2}(1) + h^{x_2}(x_5) = 2 + 2 + 11 = 15$
        - $f = g + h = 14 + 15 = 29$
    - $x_5 = 1$
        - $g = 8 + F_5(1) + F_{56}(1,0) = 8 + 2 + 0 = 10$
        - $h = h^{x_3}(1) + h^{x_2}(1) + h^{x_2}(x_5) = 2 + 2 + 0 = 4$
        - $f = g + h = 10 + 4 = 14$

  Έχουμε $f = 29 < L$. Οπισθοδρομούμε και άλλο, $x_6 = 1$.
- x_1
    - $x_1 = 0$
        - $g = 4 + F_1(0) = 4 + 0 = 4$
        - $h = h^{x_5}(1) + h^{x_3}(x_1) + h^{x_2}(x_1) = 12 + 10 + 10 = 32$
        - $f = g + h = 4 + 32 = 36$
    - $x_1 = 1$
        - $g = 4 + F_1(1) = 4 + 8 = 12$
        - $h = h^{x_5}(1) + h^{x_3}(x_1) + h^{x_2}(x_1) = 12 + 2 + 2 = 16$
        - $f = g + h = 12 + 16 = 28$

  Άρα επιλέγουμε $x_1 = 0$.
- x_5
    - $x_5 = 0$
        - $g = 4 + F_5(0) + F_{56}(0,1) = 4 + 0 + 0 = 4$
        - $h = h^{x_3}(0) + h^{x_2}(0) + h^{x_2}(x_5) = 10 + 10 + 11 = 31$
        - $f = g + h = 4 + 31 = 35$
    - $x_5 = 1$
        - $g = 4 + F_5(1) + F_{56}(1,1) = 4 + 2 + 10 = 16$
        - $h = h^{x_3}(0) + h^{x_2}(0) + h^{x_2}(x_5) = 10 + 10 + 0 = 20$
        - $f = g + h = 16 + 20 = 36$

  Άρα επιλέγουμε $x_5 = 1$.
- x_2, λόγω του R_{25} επιλέγουμε $x_2 = 0$.
    - $x_2 = 0$
        - $g = 16 + F_{12}(0,0) + F_2(0) = 16 + 8 + 0 = 24$
        - $h = h^{x_3}(0) + h^{x_4}(0, x_2) + h^{x_3}(x_2, 1) = 10 + 2 + 0 = 12$
        - $f = g + h = 24 + 12 = 36$

  Άρα συνεχίζουμε.
- x_3, λόγω του R_{35} επιλέγουμε $x_3 = 0$.
    - $x_3 = 0$
        - $g = 24 + F_{13}(0,0) + F_{235}(0,0,1) + F_3(0) = 24 + 10 + 0 + 0 = 34$
        - $h = h^{x_4}(0,0) = 2$
        - $f = g + h = 34 + 2 = 36$

  Άρα επιλέγουμε $x_3 = 0$.
- x_4
    - $x_4 = 0$
        - $g = 34 + F_{124}(0,0,0) + F_4(0) = 34 + 0 + 0 = 34$
        - $h = 0$
        - $f = g + h = 34 + 0 = 34$
    - $x_4 = 1$
        - $g = 34 + F_{124}(0,0,1) + F_4(1) = 34 + 0 + 2 = 36$
        - $h = 0$
        - $f = g + h = 36 + 0 = 36$

  Άρα επιλέγουμε $x_4 = 1$. Βρήκαμε $f = 36 > L$. Αλλάζουμε την τιμή του L σε 36.

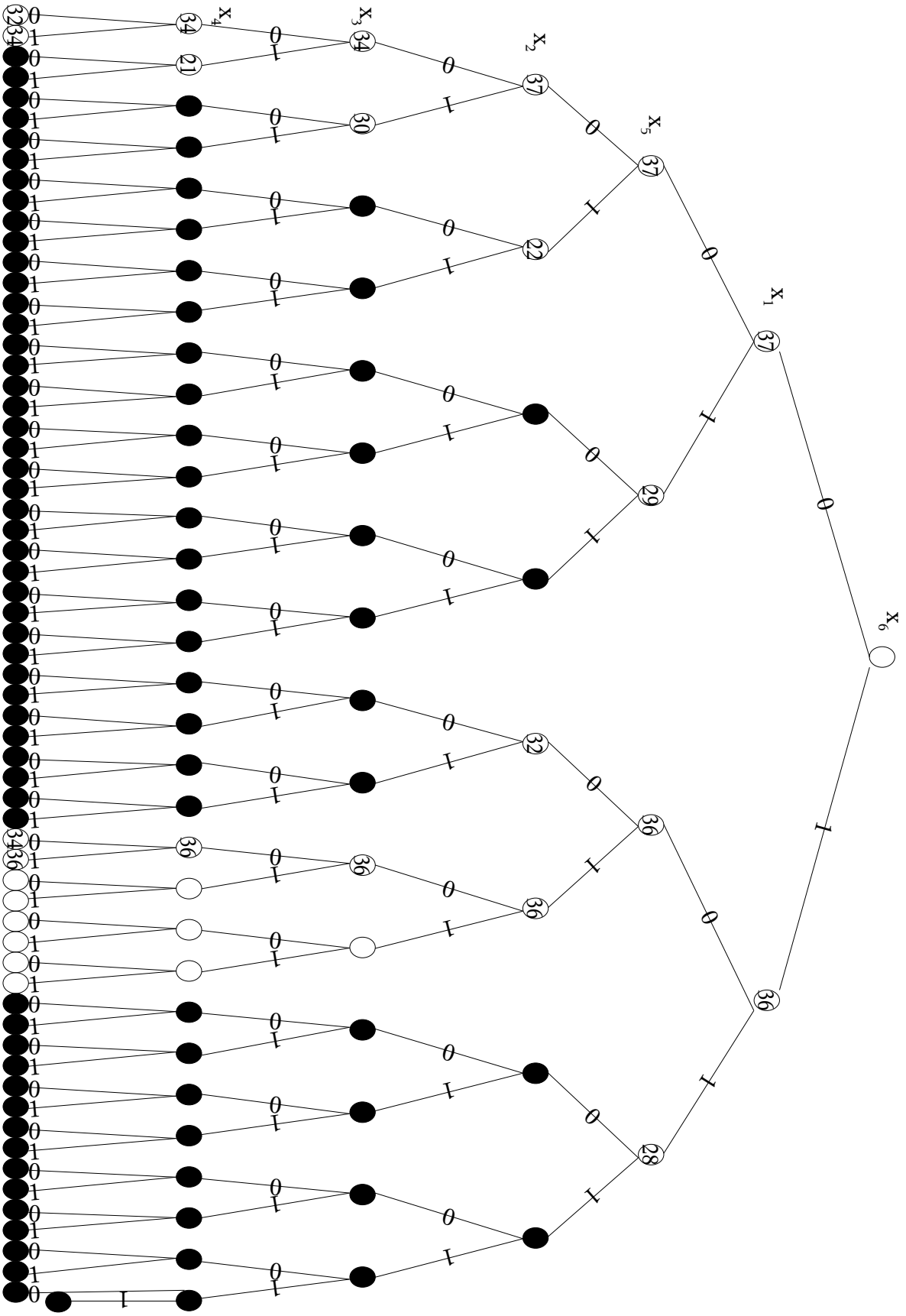
Οπισθοδρομούμε για $x_5 = 0$.

- x_2
  - $x_2 = 0$
    - $g = 4 + F_{12}(0,0) + F_2(0) = 4 + 8 + 0 = 12$
    - $h = h^{x_3}(0) + h^{x_4}(0, x_2) + h^{x_3}(x_2, 0) = 10 + 2 + 8 = 20$
    - $f = g + h = 12 + 20 = 32$
  - $x_2 = 1$
    - $g = 4 + F_{12}(0,1) + F_2(1) = 4 + 0 + 6 = 10$
    - $h = h^{x_3}(0) + h^{x_4}(0, x_2) + h^{x_3}(x_2, 0) = 10 + 0 + 5 = 15$
    - $f = g + h = 10 + 15 = 25$

  Έχουμε $f = 32 < L$. Οπισθοδρομούμε και άλλο, $x_1 = 1$.
- x_5
  - $x_5 = 0$
    - $g = 12 + F_5(0) + F_{56}(0,1) = 12 + 0 + 0 = 12$
    - $h = h^{x_3}(1) + h^{x_2}(1) + h^{x_2}(x_5) = 2 + 2 + 11 = 15$
    - $f = g + h = 12 + 15 = 27$
  - $x_5 = 1$
    - $g = 12 + F_5(1) + F_{56}(1,1) = 12 + 2 + 10 = 24$
    - $h = h^{x_3}(1) + h^{x_2}(1) + h^{x_2}(x_5) = 2 + 2 + 0 = 4$
    - $f = g + h = 24 + 4 = 28$

  Έχουμε $f = 28 < L$. Δε μπορούμε να οπισθοδρομήσουμε και άλλο. Φτάσαμε στη βέλτιστη λύση.

Η όλη διαδικασία φαίνεται καλύτερα, σχηματικά, αν ζωγραφίσουμε το δέντρο αναζήτησης. Το παραθέτουμε ευθύς αμέσως. Απολαύστε το! Φαίνεται ξεκάθαρα η απόδοση της BnB με ευριστική MBE. Επισκεπτόμαστε πολύ λίγους κόμβους.



Σχήμα 10: Το δέντρο αναζήτησης για BnB. Με μαύρο γέμισμα αναπαρίστανται οι κόμβοι που δεν επισκεφθήκαμε λόγω της BnB και με άσπρο αυτοί που δεν επισκεφθήκαμε λόγω των ισχυρών περιορισμών.

6.4 Βιβλιογραφικές αναφορές

Ένας πολύ καλός αλγόριθμος, για καλές συναρτήσεις εκτίμησης οριοθέτησης, είναι ο αλγόριθμος αναζήτησης της Μπαμπούσκα (the Russian Doll Search) [VLS96] και [Rin03].

Σημαντικό ρόλο, έχουν παίξει οι BE στα δίκτυα του Bayes (Bayesian networks ή belief networks). Για αυτά τα δίκτυα οι BE υπολογίζουν:

- Την πιο πιθανή εξήγηση (most probable explanation)[Rina99], [KR94], [RR97], [Rin99].
- Την μετέπειτα μέγιστη υπόθεση (maximum a posteriori hypothesis) [RR97], [Rin99], [Rina99].
- Εκτίμηση και ανανέωση προσδοκίας (belief assessment and updating) [RR97], [Rin99], [Rina99].
- Τη μεγιστοποίηση της προσδοκώμενης ωφέλειας (maximization of expected utility) [Rin99], [Rina99].

Οι BE έχουν βοηθήσει στο πρόβλημα του μέγιστου-CSP (max-CSP) [KR94] και στο πρόβλημα της αυτοματοποιημένης συλλογιστικής (Automated Reasoning Problem) [KR94].

Άλλες δομές που έχουν ενισχύσει ή αντικαταστήσει σε διάφορα προβλήματα, είναι η απαλοιφή με συστάδες δέντρων (cluster tree elimination) [RKL01], η απαλοιφή με μικρο-συστάδες δέντρων (mini-cluster tree elimination) [RKL01], τα πολυδένδρα (poly-trees) [Rin99], [Rina99], η συστάδα δέντρων συνένωσης (join tree clustering) [Rin99], [Rina99], οι γράφοι συνένωσης (join graphs) [RKM02] και η αποσύνθεση δέντρου (tree decomposition) [KRLC01].

Ένα πολύ γνωστό πρόβλημα είναι το παιχνίδι της ζωής (still life) και αυτό προσεγγίζεται από BE [LM03].

Ένα ενδιαφέρον δίκτυο με κόστη είναι το δίκτυο με πολλαπλά κόστη (multiobjective weighted CSP) [RELJ06]. Άλλα ενδιαφέροντα δίκτυα, είναι τα μεικτά δίκτυα (mixed networks) [RM08], μεικτά ως προς το ότι είναι και ντετερμινιστικά και πιθανοτικά (deterministic and probabilistic) ταυτόχρονα!

Άλλα προβλήματα που αντιμετωπίζονται με BE, είναι οι γραμμικές ανισότητες (linear inequalities) [Rina99] και η κατασκευή τυχαίων λύσεων (generating random solutions) [RKBE02].

Μία πολύ καλή εισαγωγή στους αλγόριθμους πάσης στιγμής [Zil96].

7 Βελτιώνοντας την απαλοιφή με μικρο-κάδους

7.1 Δέντρο υπολογισμού

Το δέντρο υπολογισμού, το συναντήσαμε στην απόδειξη της πολυπλοκότητας του MBE-OPT-CONS(i). Θα σχεδιάσουμε το δέντρο υπολογισμού για το προηγούμενο παράδειγμα, για να το κατανοήσουμε καλύτερα. Αναφέρουμε τι είναι δέντρο υπολογισμού.

Ορισμός: Δέντρο υπολογισμού

Δέντρο υπολογισμού (computation tree) είναι ένα δέντρο που αναφέρεται σε μία σειρά από υπολογισμούς (computations), με τις εξής ιδιότητες:

- *Στα φύλλα του δέντρου υπάρχουν οι αρχικοί υπολογισμοί.*
- *Στους εσωτερικούς κόμβους του δέντρου έχουμε υπολογισμούς που είναι σύνθεση των παιδιών τους.*
- *Η ρίζα του δέντρου είναι το αποτέλεσμα του υπολογισμού.*

7.1.1 Παράδειγμα δέντρου υπολογισμού

Θα χρησιμοποιήσουμε το παράδειγμά για MBE-OPT-CONS(i). Χάριν ευκολίας παραθέτουμε αυτό που μας ενδιαφέρει από το συγκεκριμένο παράδειγμα, την τελική μορφή των κάδων.

Κάδος Συναρτήσεις

Bucket(x_4): $F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $R_{24}(x_2,x_4)$, $R_{14}(x_1,x_4)$

Bucket(x_3): $F_3(x_3)$, $F_{235}(x_2,x_3,x_5)$, $F_{13}(x_1,x_3)$, $R_{35}(x_3,x_5)$, $R_{13}(x_1,x_3)$

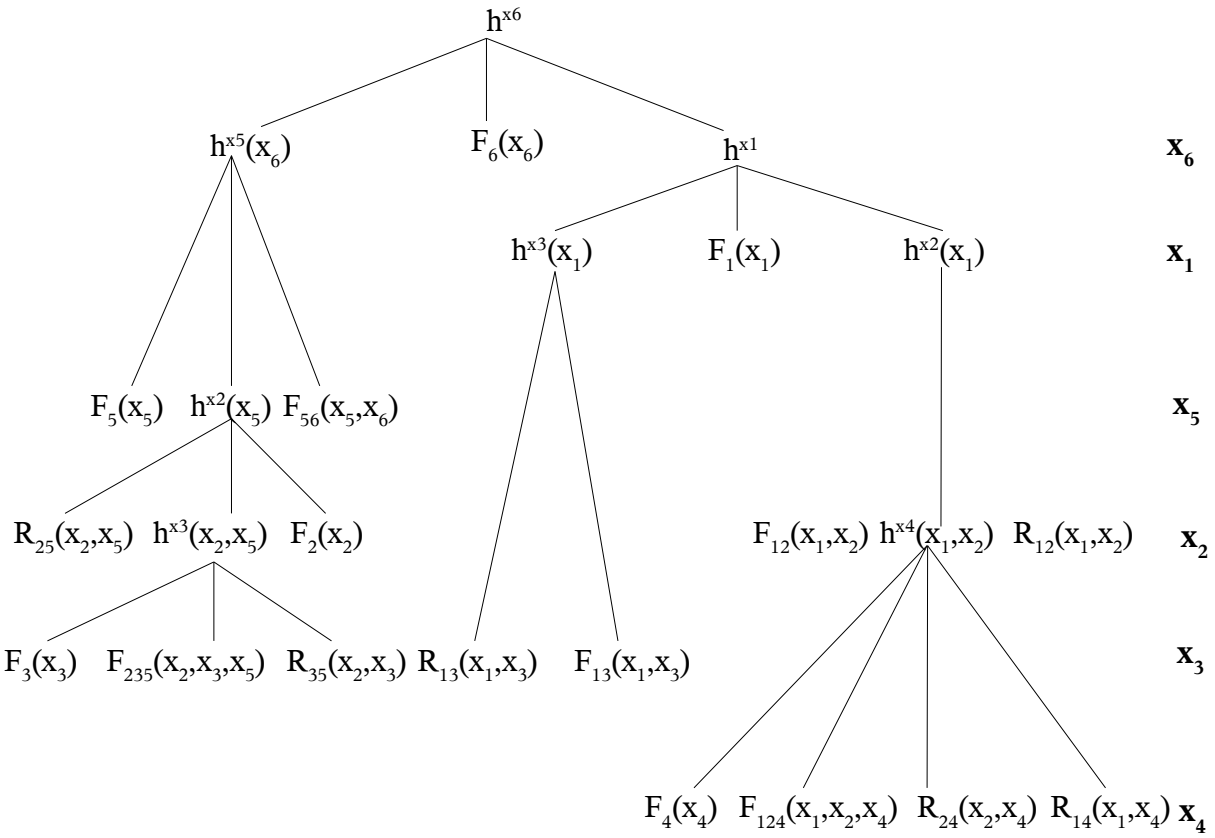
Bucket(x_2): $F_2(x_2)$, $F_{12}(x_1,x_2)$, $R_{12}(x_1,x_2)$, $R_{25}(x_2,x_5)$, $h^{x_4}(x_1,x_2)$, $h^{x_3}(x_2,x_5)$

Bucket(x_5): $F_5(x_5)$, $F_{56}(x_5,x_6)$, $h^{x_2}(x_5)$

Bucket(x_1): $F_1(x_1)$, $h^{x_3}(x_1)$, $h^{x_2}(x_1)$

Bucket(x_6): $F_6(x_6)$, $h^{x_5}(x_6)$, h^{x_1}

Ας δούμε το δέντρο υπολογισμού. Στα φύλλα υπάρχουν οι αρχικές συναρτήσεις – περιορισμοί. Στους εσωτερικούς κόμβους του δέντρου έχουμε συναρτήσεις που έχουν δημιουργηθεί από τα παιδιά τους με απαλοιφή μεταβλητών. Η ρίζα αυτού του δέντρου είναι το προσεγγισμένο βέλτιστο κόστος του πρώτου κάδου. Για όλους τους κόμβους του δέντρου, ο υπολογισμός που γίνεται είναι ο $h^{mb}(t)= \max_{a_p \text{έ.ώ.} R^{x_i}(t,a_p)} \{\sum_{i=1}^{j} h^{x_i}(t,a_p)+\sum_{i=1} F_{Q_i}(t,a_p)\}$, με $R^{mb}=\pi_{U_{mb}} \text{JOINT}_{i=1}^{k} R_i$.
Ακολουθεί το δέντρο υπολογισμού:



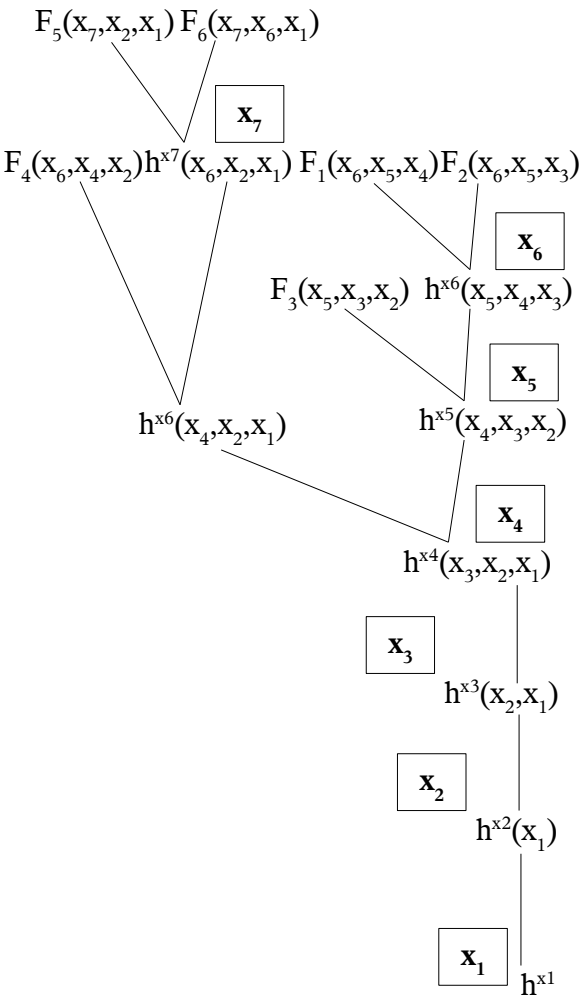
Σχήμα 11: Δέντρο υπολογισμού

7.1.2 Δέντρα υπολογισμού και απαλοιφή με μικρο-κάδους

Τα δέντρα υπολογισμού δεν είναι μόνο χρήσιμα για την απόδειξη της πολυπλοκότητας του MBE-OPT-CONS(i). Είναι πολύ χρήσιμα για τη βελτίωση της χωρικής πολυπλοκότητας των MBE. Θα δούμε διάφορους τρόπους με τους οποίους μπορούμε να βελτιώσουμε τη χωρική πολυπλοκότητα του προηγούμενου παραδείγματος!

Αν και είναι πολύ δύσκολο να βρούμε το αποτέλεσμα ενός MBE αλγορίθμου, είναι πολύ εύκολο να γνωρίζουμε από την αρχή την τελική μορφή των κάδων του. Αυτό το έχουμε κάνει σε όλα τα παραδείγματα που έχουμε παραθέσει έως τώρα. Γνωρίζουμε εκ των προτέρων την τελική μορφή των κάδων, απλά δεν έχουμε κάνει τους υπολογισμούς ώστε να πάρουμε και το τελικό αποτέλεσμα. Αυτή η ιδιότητα των MBE μας βοηθά ώστε να μπορούμε να αναδιοργανώσουμε τους κάδους και να επιτύχουμε καλύτερη χωρική πολυπλοκότητα.

Τις τεχνικές, που θα αναφέρουμε, μπορούμε να τις χρησιμοποιήσουμε σε οποιοδήποτε δέντρο υπολογισμού. Ωστόσο, δε θα χρησιμοποιήσουμε το δικό μας γιατί δεν προσφέρεται για όλες τις μεθόδους που θα αναφέρουμε και παραθέτουμε ένα άλλο «στημένο» δέντρο.



Σχήμα 12: Το δέντρο υπολογισμού. Στα τετράγωνα πλαίσια εμφανίζεται η μεταβλητή που μόλις έχει απαλειφθεί.

Πριν ξεκινήσουμε να αναφέρουμε τις τεχνικές με τις οποίες θα καταφέρουμε να βελτιώσουμε τη χωρική πολυπλοκότητα των ΜΒΕ, θα υπολογίσουμε το κόστος σε χώρο του δέντρου υπολογισμού. Πρέπει να αναφέρουμε ότι στη μελέτη για τη χωρική πολυπλοκότητα δεν παίρνουμε υπόψιν μας τις αρχικές συναρτήσεις – περιορισμούς F_i, καθώς αυτές δεν μπορούμε να τις αποφύγουμε, αποτελούν την είσοδο των ΜΒΕ.

Έχουμε πέντε συναρτήσεις με εμβέλεια τριών μεταβλητών, μία συνάρτηση με εμβέλεια δύο μεταβλητών, άλλη μία συνάρτηση με εμβέλεια μιας μεταβλητής και μια συνάρτηση χωρίς μεταβλητές. Ας υποθέσουμε ότι τις συναρτήσεις τις αποθηκεύουμε σε πίνακες και ότι το πεδίο τιμών των συναρτήσεων είναι ένα δισύνολο, (όπως ακριβώς των δημοπρασιών). Το πλήθος των στοιχείων των πινάκων που χρειάζεται να αποθηκεύσουμε όλες αυτές τις συναρτήσεις είναι:
$5 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 47$. Ας δούμε πως θα βελτιώσουμε το χωρικό κόστος.

7.1.2.1 Αναδιάταξη κλαδιών

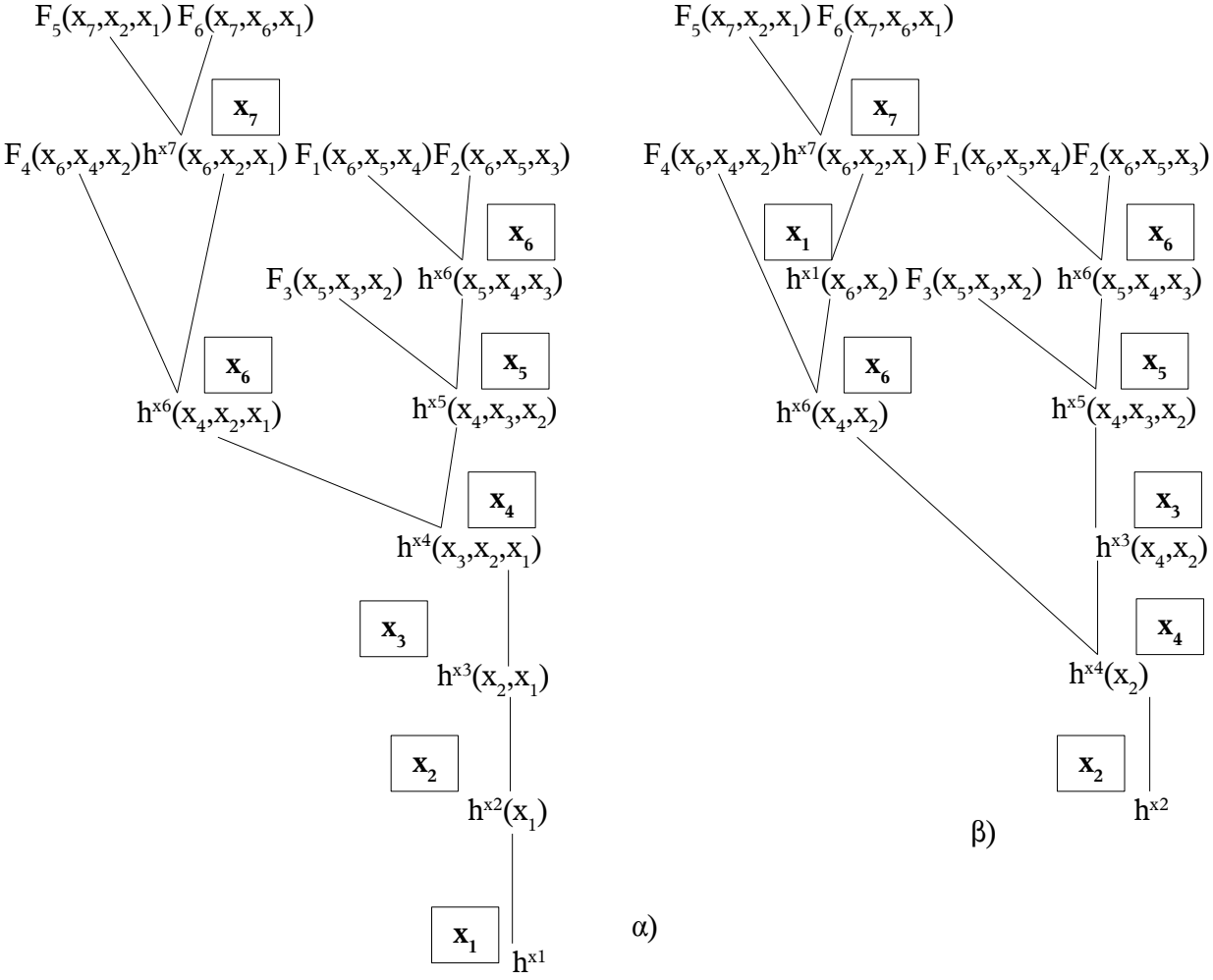
Η πρώτη τεχνική με την οποία θα γλιτώσουμε χώρο καλείται «αναδιάταξη κλαδιών» (branch rearrangement). Στο δέντρο υπολογισμού, μπορούμε να παρατηρήσουμε ότι η μεταβλητή x_1 απαλείφεται τελευταία, ενώ συμμετέχει πολύ νωρίς στον υπολογισμό. Αυτό συνεπάγεται ότι όλες οι συναρτήσεις σχεδόν θα περιέχουν στην εμβέλεια τους τη x_1 και θα επιβαρύνουν χωρίς λόγο τη χωρική πολυπλοκότητα. Αν μπορούσαμε να απαλείψουμε νωρίς τη x_1, θα γλιτώναμε πολύ χώρο. Αυτό γίνεται, αλλά όχι πάντα. Ας δούμε έναν κανόνα με τον οποίο μπορούμε να δούμε πότε

μπορούμε να κάνουμε κάτι τέτοιο:

Κανόνας αναδιάταξης κλαδιών:

> *Έστω ένας κόμβος v στο δέντρο υπολογισμού, τέτοιος που να έχει ένα παιδί, και έστω x_v η μεταβλητή που έχει απαλειφθεί στο v. Έστω u ο πρώτος απόγονος του v που έχει πάνω από ένα παιδί. Εάν μόνο ένα από τα παιδιά του v έχει στην εμβέλεια τη x_v, τότε ο κόμβος v μπορεί να μετακινηθεί μεταξύ u και w, όπου w το μοναδικό παιδί του u που έχει στην εμβέλεια του τη x_v.*

Ο κανόνας αυτός εκτελείται συνεχώς για όλους τους κόμβους v του δέντρου. Σημειωτέον, εάν ο κανόνας επιφέρει αλλαγές σε ένα κλαδί για ένα κόμβο v, τότε πρέπει να ξανατρέξει για τον κόμβο αυτό, όπως ακριβώς γίνεται για τη x_1. Αρχικά, v = h^{x_1} , u = $h^{x_4}(x_3,x_2,x_1)$, w = $h^{x_6}(x_4,x_2,x_1)$. Μετά ο κανόνας θα εκτελεσθεί ξανά, θα έχουμε νέο u και w για τη x_1 και θα καταλήξουμε στο εξής δέντρο υπολογισμού (θα γίνουν αλλαγές και για τη x_3).



Σχήμα 13: Το δέντρο υπολογισμού πριν α) και μετά β) την αναδιάταξη κλαδιών

Το χωρικό κόστος για το νέο δέντρο υπολογισμού είναι: $3\cdot2^3 + 3\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 39$! Απίστευτο. Για να δούμε τι καλύτερο μπορούμε να κάνουμε.

7.1.2.2 Κάθετη σύμπτυξη

Παρατηρούμε ότι στο νέο δέντρο υπολογισμού, ο υπολογισμός της h^{x_2} από τη $h^{x_4}(x_2)$ μπορεί να γίνει σε ένα βήμα παρόλο που στο MBE γίνεται σε δύο βήματα, καθώς οι δύο

συναρτήσεις βρίσκονται σε διαφορετικούς κάδους. Με το να γίνει σε ένα βήμα ο υπολογισμός της θα κερδίσουμε χώρο αφού δε θα χρειαστεί να αποθηκεύσουμε τη $h^{x_4}(x_2)$. Όπως και πριν, δε συμμετέχουν στην όλη διαδικασία οι αρχικές συναρτήσεις – περιορισμοί. Την τεχνική, την ονομάζουμε κάθετη σύμπτυξη (vertical compaction). Πριν δώσουμε τον κανόνα της κάθετη σύμπτυξης ας δούμε έναν ορισμό.

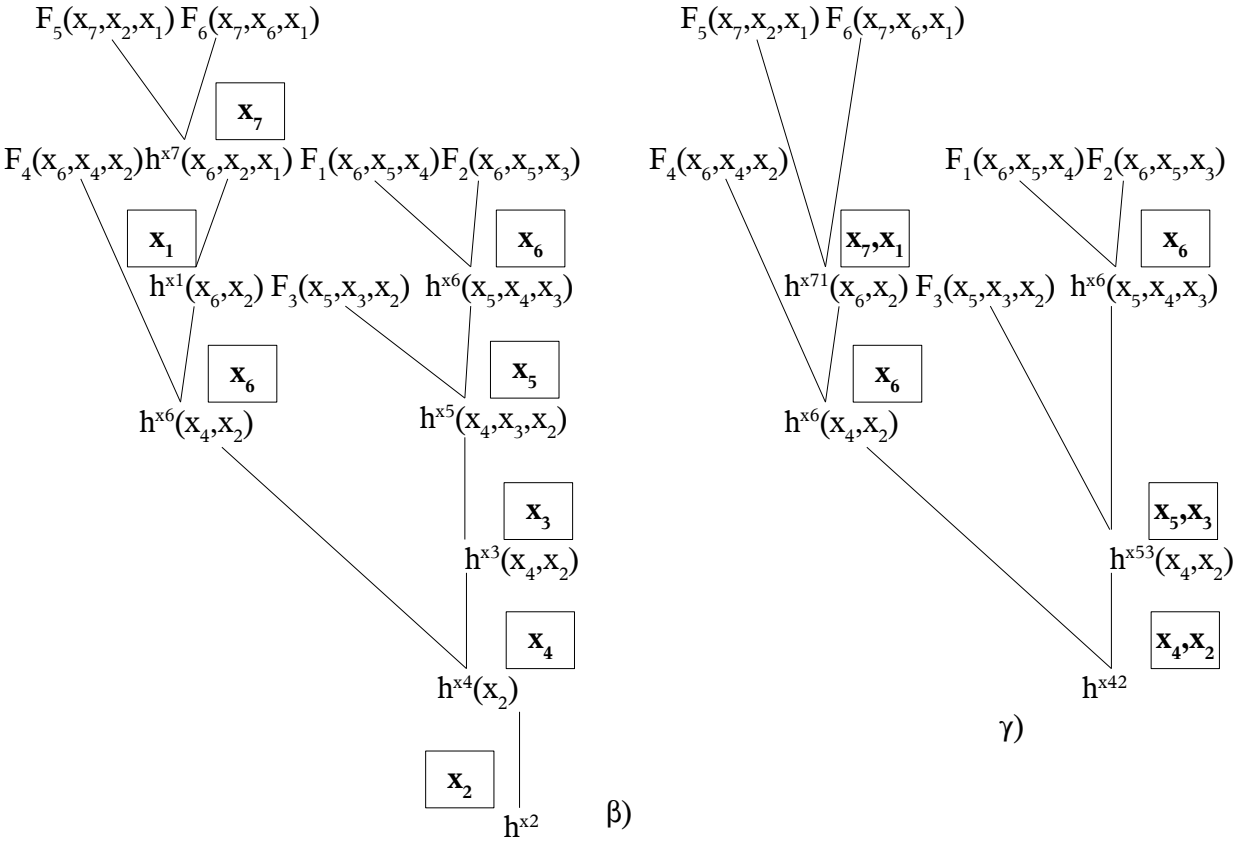
Ορισμός: Εσωτερικό γραμμικό μονοπάτι

Το μονοπάτι στο δέντρο, στο οποίο συμμετέχουν κόμβοι χωρίς παιδιά το ονομάζουμε εσωτερικό γραμμικό μονοπάτι (internal linear path).

Κανόνας κάθετης σύμπτυξης

Συγχωνεύουμε όλα τα εσωτερικά γραμμικά μονοπάτια του δέντρου σε έναν κόμβο.

Στο παράδειγμά μας, συγχωνεύουμε τα εσωτερικά γραμμικά μονοπάτια $(h^{x_2}, h^{x_4}(x_2))$, $(h^{x_4}(x_4,x_2), h^{x_5}(x_4,x_3,x_2))$ και $(h^{x_1}(x_6,x_2), h^{x_7}(x_6,x_2,x_1))$ και παίρνουμε τις $h^{x_{42}}$, $h^{x_{53}}(x_4,x_2)$ και $h^{x_{71}}(x_6,x_2)$ αντίστοιχα. Το δέντρο που προκύπτει αναπαριστάται στο επόμενο σχήμα.



Σχήμα 14: Το δέντρο υπολογισμού πριν β) και μετά γ) την κάθετη σύμπτυξη

Το χωρικό κόστος για το νέο δέντρο υπολογισμού είναι: $1 \cdot 2^3 + 3 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21$! Θα δοκιμάσουμε να το ελαττώσουμε περισσότερο!

7.1.2.3 Οριζόντια σύμπτυξη

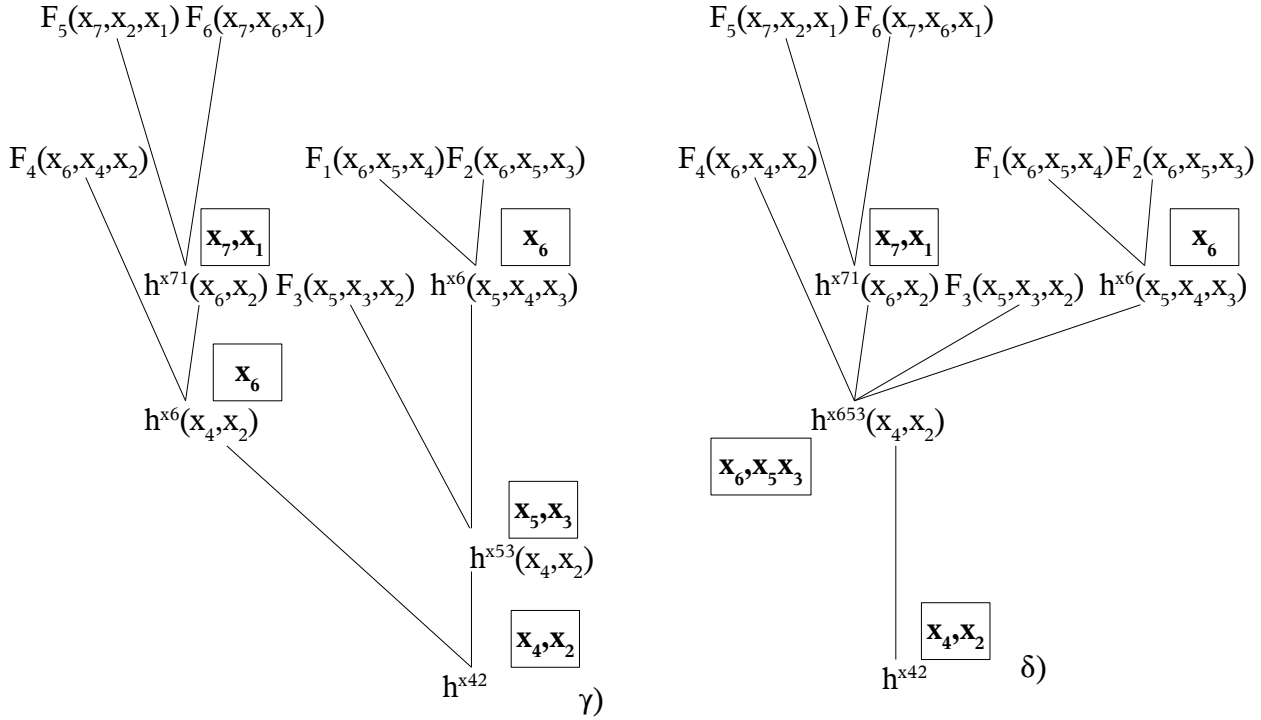
Παρατηρήστε στο καινούργιο δέντρο τις $h^{x_6}(x_4,x_2)$ και $h^{x_3}(x_4,x_2)$. Εάν τις

υπολογίσουμε παράλληλα, μπορούμε να γλιτώσουμε παραπάνω χώρο. Αυτό το είδος σύμπτυξης ονομάζεται οριζόντια σύμπτυξη (horizontal compaction).

Κανόνας οριζόντιας σύμπτυξης

Εάν δύο κόμβοι είναι ορισμένοι στο ίδιο σύνολο μεταβλητών τότε τους υπολογίζουμε παράλληλα.

Ο νέος κόμβος είναι ο $h^{x_{653}}(x_4, x_2)$. Το δέντρο που προκύπτει αναπαρίσταται στο επόμενο σχήμα.



Σχήμα 15: Το δέντρο υπολογισμού πριν γ) και μετά δ) την οριζόντια σύμπτυξη

Το χωρικό κόστος για το νέο δέντρο υπολογισμού είναι: $1 \cdot 2^3 + 2 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 17$! Καταφέραμε να ελαττώσουμε το χωρικό κόστος από 47 σε 17, 64% μείωση!

7.1.2.4 MBE πρώτα κατά βάθος

Υπάρχει και ένας ακόμα τρόπος να ελαττώσουμε τη χωρική πολυπλοκότητα των MBE. Αυτή τη φορά δεν επεμβαίνουμε στη δομή του δέντρου υπολογισμού, αλλά στο πως ο MBE διασχίζει το δέντρο και στο τι κρατάει στη μνήμη από τους υπολογισμούς που κάνει.

Ας θυμηθούμε πως λειτουργεί ο MBE. Υπολογίζει τα κόστη σε όλους τους κόμβους του δέντρου, ώστε αργότερα να επιστρέψει και να υπολογίσει το αριστερό άκρο του διαστήματος που υπολογίζει τη συνάρτηση κόστους. Μπορούμε να θυσιάσουμε το αριστερό άκρο (κάτω όριο) και να μην κρατάμε όλα τα κόστη που έχουμε υπολογίσει[9].

Για κάθε κόμβο που υπολογίζουμε το κόστος του, όταν το υπολογίσουμε, διαγράφουμε από τη μνήμη τα παιδιά του. Μια τέτοια ενέργεια είναι πολύ εύκολο να υλοποιηθεί εάν διασχίσουμε το δέντρο με την πρώτα κατά βάθος αναζήτηση (depth first search). Χάριν συντομίας, θα αναφερόμαστε στην πρώτα κατά βάθος αναζήτηση με το ακρωνύμιο DFS που παράγεται από τους αγγλικούς όρους depth first search.

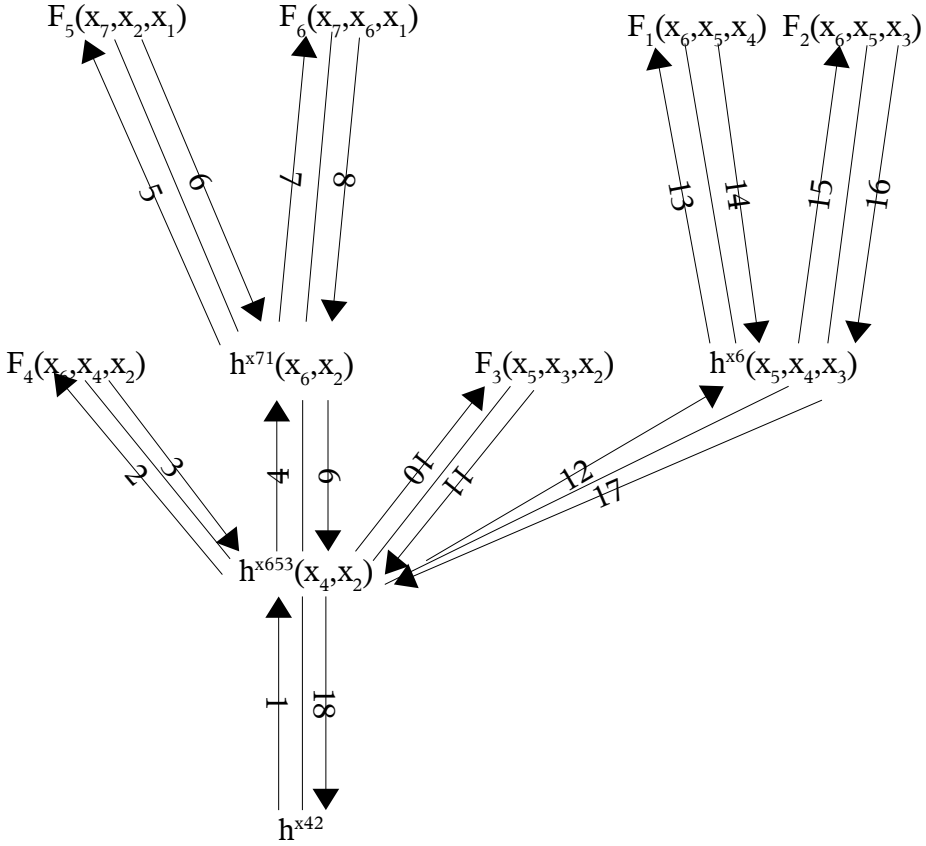
Πόσο θα είναι η χωρική πολυπλοκότητα; Θα την υπολογίσουμε αναδρομικά. Έστω ότι είμαστε σε ένα κόμβο v και θέλουμε να υπολογίσουμε τη χωρική πολυπλοκότητα για αυτόν τον

9 Μπορούμε να γλιτώσουμε τη θυσία του κάτω ορίου της συνάρτησης κόστους κρατώντας τα ενδιάμεσα αποτελέσματα, που υπολογίζουμε στους κόμβους, στο σκληρό δίσκο.

κόμβο. Αυτή θα είναι ίση με το μέγιστο από τα επόμενα μεγέθη (να θυμάστε πάντα ότι τρέχουνε πρώτα κατά βάθος αναζήτηση):

- Τη μνήμη που χρειάζεται ο κόμβος v και τα παιδιά του.
- Ας θεωρήσουμε μια απαρίθμηση των παιδιών του $w_1,...,w_k$, όπου k το πλήθος των παιδιών του. Για κάθε παιδί w_i υπολογίζουμε το άθροισμα των χωρικών πολυπλοκοτήτων των παιδιών w_j με $j < i$ και προσθέτουμε τη μνήμη που σπαταλά το παιδί αυτό.

Για να καταλάβετε γιατί ισχύει η παραπάνω ανάλυση, πρέπει να φανταστείτε στο μυαλό σας τη DFS στο δέντρο υπολογισμού. Για να σας βοηθήσω, θα σας δώσω την κατάσταση της μνήμης κάθε φορά που ο πρώτα κατά βάθος MBE επισκέπτεται έναν κόμβο.



Σχήμα 16: Η πρώτα κατά βάθος αναζήτηση στο δέντρο υπολογισμού

Στο παραπάνω σχήμα, φαίνεται η DFS στο δέντρο υπολογισμού. Κάθε επίσκεψη της αναζήτησης έχει ζωγραφιστεί με ένα βελάκι. Ορίστε και η κατάσταση της μνήμης ανάλογα με το βήμα στο οποίο βρίσκεται η DFS.

Επισκέψεις	Επισκεπτόμενος Κόμβος	Μνήμη
Επίσκεψη 0	$h^{x_{42}}$	κενή
Επίσκεψη 1	$h^{x_{653}}$	κενή
Επίσκεψη 2	F_4	F_4
Επίσκεψη 3	$h^{x_{653}}$	F_4
Επίσκεψη 4	$h^{x_{71}}$	F_4
Επίσκεψη 5	F_5	F_4, F_5
Επίσκεψη 6	$h^{x_{71}}$	F_4, F_5
Επίσκεψη 7	F_6	F_4, F_5, F_6
Επίσκεψη 8	$h^{x_{71}}$	F_4, $h^{x_{71}}$
Επίσκεψη 9	$h^{x_{653}}$	F_4, $h^{x_{71}}$
Επίσκεψη 10	F_3	F_4, $h^{x_{71}}$, F_3
Επίσκεψη 11	$h^{x_{653}}$	F_4, $h^{x_{71}}$, F_3
Επίσκεψη 12	h^{x_6}	F_4, $h^{x_{71}}$, F_3
Επίσκεψη 13	F_1	F_4, $h^{x_{71}}$, F_3, F_1
Επίσκεψη 14	h^{x_6}	F_4, $h^{x_{71}}$, F_3, F_1
Επίσκεψη 15	F_2	F_4, $h^{x_{71}}$, F_3, F_1, F_2
Επίσκεψη 16	h^{x_6}	F_4, $h^{x_{71}}$, F_3, h^{x_6}
Επίσκεψη 17	$h^{x_{653}}$	$h^{x_{653}}$
Επίσκεψη 18	$h^{x_{42}}$	$h^{x_{42}}$

Πίνακας 13: Η κατάσταση της μνήμης του πίνακα κατά τη DFS

Δεν έχουμε δώσει τον αναδρομικό τύπο. Αν με v συμβολίζεται ο κόμβος του οποίου ψάχνουμε τη χωρική πολυπλοκότητα R(v), με sp(v) η μνήμη που καταλαμβάνει ο κόμβος v, και με $(w_1,...,w_k)$, έχουμε:

$$R(v)=\max_{i=1}^{k+1}\left\{R(w_i)+\sum_{j=1}^{i-1}sp(w_j)\right\}$$, με sp()=0 και $R(w_{k+1})$ = sp(v).

Ας εφαρμόσουμε τον τύπο για το παράδειγμά μας. Παραλείπουμε τις αρχικές συναρτήσεις περιορισμούς γιατί είναι μέρος της εισόδου.

$$R(h^{x_{42}})=$$
$$=\max\{R(h^{x_{653}},sp(h^{x_{42}})+sp(h^{x_{653}}))\}=$$
$$=\max\left\{\max\{R(h^{x_{71}}),R(h^{x_6})+sp(h^{x_{71}}),sp(h^{x_{653}})+sp(h^{x_6})+sp(h^{x_{71}})\},sp(h^{x_{42}})+sp(h^{x_{653}})\right\}=$$
$$=\max\left\{\max\{sp(h^{x_{71}}),sp(h^{x_6})+sp(h^{x_{71}}),sp(h^{x_{653}})+sp(h^{x_6})+sp(h^{x_{71}})\},sp(h^{x_{42}})+sp(h^{x_{653}})\right\}=$$
$$=\max\left\{\max\{4,8+4,4+8+4\},1+4\right\}=16$$

Γλυτώσαμε λίγο ακόμα! Τέλος, ναι μπορούμε να βελτιστοποιήσουμε και άλλο το χωρικό κόστος. Ο αλγόριθμος ορίζει μία απαρίθμηση των παιδιών ενός κόμβου. Για διαφορετικές απαριθμήσεις λαβαίνουμε διαφορετικά κόστη. Αυτή τη βελτιστοποίηση μπορούμε να τη δούμε, εάν τρέξουμε DFS MBE για το δέντρο υπολογισμού «γ)», που έχει προκύψει μετά την κάθετη σύμπτυξη.

Ας δούμε τι εννοούμε. Πρώτα θα δώσουμε το χωρικό κόστος με τη DFS MBE, εάν πάρουμε πρώτα τον κόμβο $h^{x_6}(x_4,x_2)$ και μετά τον κόμβο $h^{x_{53}}(x_4,x_2)$.

$$R(h^{x_{42}})=$$
$$=\max\{R(h_2^{x_6}),R(h^{x_{53}})+sp(h^{x_6}),sp(h^{x_6})+sp(h^{x_{53}})+sp(h^{x_{42}})\}=$$ [10]
$$=\max\{\max\{R(h^{x_{71}}),sp(h^{x_{71}})+sp(h^{x_{53}})\},\max\{R(h_3^{x_6}),sp(h_3^{x_6})+sp(h^{x_{53}})\}+sp(h_2^{x_6})$$

10 Επειδή έχουμε την h^{x_6} ορισμένη σε δύο και σε τρεις μεταβλητές, αναφερόμαστε με τον δείκτη 2, $h_2^{x_6}$,όταν αναφερόμαστε σε δύο μεταβλητές και με τον δείκτη 3, $h_3^{x_6}$,όταν αναφερόμαστε σε τρεις.

$$,\mathrm{sp}(h_2^{x_6})+\mathrm{sp}(h^{x_{53}})+\mathrm{sp}(h^{x_{42}})\}=$$
$$=\max\{\max\{\mathrm{sp}(h^{x_{71}}),\mathrm{sp}(h^{x_{71}})+\mathrm{sp}(h^{x_{53}})\},\max\{\mathrm{sp}(h_3^{x_6}),\mathrm{sp}(h_3^{x_6})+\mathrm{sp}(h^{x_{53}})\}+\mathrm{sp}(h_2^{x_6})$$
$$,\mathrm{sp}(h_2^{x_6})+\mathrm{sp}(h^{x_{53}})+\mathrm{sp}(h^{x_{42}})\}=$$
$$=\max\{\max\{4,4+4\},\max\{8,8+4\}+4,4+4+1\}=16$$

Θα δώσουμε το χωρικό κόστος με τη DFS MBE, εάν πάρουμε πρώτα τον κόμβο $h^{x_{53}}(x_4,x_2)$ και μετά τον κόμβο $h^{x_6}(x_4,x_2)$.

$$R(h^{x_{42}})=$$
$$=\max\{R(h^{x_{53}}),R(h_2^{x_6})+\mathrm{sp}(h^{x_{53}}),\mathrm{sp}(h^{x_6})+\mathrm{sp}(h^{x_{53}})+\mathrm{sp}(h^{x_{42}})\}=$$
$$=\max\{\max\{R(h_3^{x_6}),\mathrm{sp}(h_3^{x_6})+\mathrm{sp}(h^{x_{53}})\},\max\{R(h^{x_{71}}),\mathrm{sp}(h^{x_{71}})+\mathrm{sp}(h^{x_{53}})\}+\mathrm{sp}(h^{x_{53}})$$
$$,\mathrm{sp}(h_2^{x_6})+\mathrm{sp}(h^{x_{53}})+\mathrm{sp}(h^{x_{42}})\}=$$
$$=\max\{\max\{\mathrm{sp}(h_3^{x_6}),\mathrm{sp}(h_3^{x_6})+\mathrm{sp}(h^{x_{53}})\},\max\{\mathrm{sp}(h^{x_{71}}),\mathrm{sp}(h^{x_{71}})+\mathrm{sp}(h^{x_{53}})\}+\mathrm{sp}(h^{x_{53}})$$
$$,\mathrm{sp}(h_2^{x_6})+\mathrm{sp}(h^{x_{53}})+\mathrm{sp}(h^{x_{42}})\}=$$
$$=\max\{\max\{8,8+4\},\max\{4,4+4\}+4,4+4+1\}=12$$

Έχουμε καταφέρει την καλύτερη έως τώρα μείωση. Ωστόσο, πρέπει πάλι να υπενθυμίσουμε ότι με DFS MBE κερδίζουμε στο χωρικό κόστος πάρα πολύ αλλά χάνουμε σε ακρίβεια, το κάτω όριο της συνάρτησης κόστους δεν προσεγγίζεται τόσο καλά.

Τα νέα είναι χαρμόσυνα και το παραπάνω πρόβλημα λύθηκε! Αντί να σβήνουμε από τη μνήμη τις συναρτήσεις που υπολογίζουμε, όταν έχει υπολογισθεί ο πατέρας τους, τις αποθηκεύουμε στο σκληρό δίσκο και, όταν αργότερα τις χρειαστούμε για να υπολογίσουμε το κάτω όριο της συνάρτησης κόστους, τις επαναφέρουμε στη μνήμη.

7.2 Ισοδύναμα δίκτυα με κόστη

Θα προσπαθήσουμε να βελτιώσουμε την επίδοση των MBE. Θα τους τροποποιήσουμε ώστε να βρίσκουν ένα καλύτερο άνω φράγμα στη συνάρτηση κόστους (υποθέτουμε ότι επιδιώκουμε τη μεγιστοποίηση της). Αυτό θα το καταφέρουμε με τη βοήθεια των ισοδύναμων δικτύων με κόστη.

Ορισμός: Ισοδύναμα δίκτυα με κόστη

> *Δύο δίκτυα με κόστη που η συνάρτηση κόστους του ενός ισούται με τη συνάρτηση κόστους του άλλου, ονομάζονται ισοδύναμα δίκτυα (equivalent networks).*

Ας δούμε ένα πολύ απλό παράδειγμα, για να καταλάβουμε την έννοια των ισοδύναμων δικτύων με κόστη. Εάν σε ένα δίκτυο με κόστη, κάθε ζεύγος συναρτήσεων κόστους το αντικαταστήσουμε με το άθροισμα τους, προκύπτει ένα ισοδύναμο δίκτυο. Ο τρόπος με τον οποίο μετασχηματίζουμε ένα δίκτυο με κόστη σε ένα ισοδύναμό του ονομάζεται μετασχηματισμός (transformation).

Ας δούμε τι θέλουμε να αλλάξουμε στο MBE. Έχουμε πει ότι στο MBE χάνουμε ως προς το άνω φράγμα της συνάρτησης κόστους, υποθέτοντας ότι αναζητάμε τη μεγιστοποίηση της. Γιατί χάνουμε; Σε κάθε κάδο η μεταβλητή που απαλείφουμε υπάρχει σε όλες τις συναρτήσεις του κάδου. Χωρίζοντας τον κάδο σε μικρο-κάδους, πληροφορία για τη μεταβλητή που απαλείφουμε μπορεί να χαθεί, καθώς δε συνενώνονται οι συναρτήσεις σε διαφορετικούς μικρο-κάδους.

Αυτό το χάσιμο της πληροφορίας, θα προσπαθήσουμε να το μειώσουμε. Η ιδέα είναι να μεταφέρουμε κόστος από όλες τις συναρτήσεις στον κάδο, σε μία μόνο. Πληροφορία που μπορεί να είχε χαθεί λόγω του διαχωρισμού του κάδου σε μικρο-κάδους, μπορεί να μη χαθεί. Θα μετασχηματίσουμε το δίκτυό μας σε ένα άλλο ισοδύναμο, που θα έχει τα κόστη συσσωρευμένα σε μία συνάρτηση σε κάθε κάδο.

7.2.1 Κανόνας μετασχηματισμού μεταφοράς κόστους.

Έστω ότι έχουμε δύο συναρτήσεις κόστους F_i και F_j. Μεταφέρουμε κόστος από τη F_i στη F_j, με τέσσερα βήματα:

i. *Βρίσκουμε την τομή των εμβελειών $var(F_i)$ και $var(F_j)$ των δύο συναρτήσεων. Με var(F) συμβολίζουμε την εμβέλεια της συνάρτησης F.*

ii. *Υπολογίζουμε τη συνάρτηση h, $h = f[var(F_i) \cap var(F_j)]$.*

iii. *Υπολογίζουμε τη συνάρτηση $F_i = F_i - h$.*

iv. *Υπολογίζουμε τη συνάρτηση $F_j = F_j + h$.*

7.2.1.1 Παράδειγμα μεταφοράς κόστους

Πριν αναφέρουμε αναλυτικά τι θα κάνουμε στον ΜΒΕ, ώστε να προσθέσουμε τη μεταφορά του κόστους μεταξύ συναρτήσεων, ας δούμε ένα παράδειγμα.
Έχουμε τις συναρτήσεις:

$$F_{13}(x_1,x_3)=\begin{cases} 8, \text{εάν } x_1=x_3=1 \\ 2, \text{εάν } x_1=1 \text{ και } x_3=0 \\ 10, \text{εάν } x_1=0 \text{ και } x_3=0 \\ 0, \text{αλλιώς} \end{cases} \quad \text{και} \quad F_{235}(x_2,x_3,x_5)=\begin{cases} 8, \text{εάν } x_2=x_3=x_5 \\ 0, \text{αλλιώς} \end{cases} .$$

Μας βολεύει να γράφουμε τις συναρτήσεις με τη μορφή πινάκων. Έχουμε:

x_1	x_3	F_{13}	x_2	x_3	x_5	F_{235}
0	0	10	0	0	0	8
0	1	0	0	0	1	0
1	0	2	0	1	0	0
1	1	8	0	1	1	0
			1	0	0	0
			1	0	1	0
			1	1	0	0
			1	1	1	8

Πίνακας 14: Οι συναρτήσεις F_{13} και F_{235} υπό τη μορφή πινάκων

Θα μεταφέρουμε κόστος από τη F_{13} στη F_{235}. Η μεταφορά κόστους μπορεί να επιτευχθεί, καθώς οι δύο συναρτήσεις έχουν κοινή μεταβλητή στην εμβέλειά τους, τη x_3. Για $x_3 = 0$, μπορούμε να μεταφέρουμε κόστος 2 και για $x_3 = 1$ δεν μπορούμε να μεταφέρουμε καθόλου κόστος. Οι συναρτήσεις θα γίνουν έως εξής:

x_1	x_3	F_{13}	x_2	x_3	x_5	F_{235}
0	0	8	0	0	0	10
0	1	0	0	0	1	2
1	0	0	0	1	0	0
1	1	8	0	1	1	0
			1	0	0	2
			1	0	1	2
			1	1	0	0
			1	1	1	10

Πίνακας 15: Οι συναρτήσεις μετά τη μεταφορά κόστους

7.2.2 Μεταφορά κόστους και απαλοιφή με μικρο-κάδους

7.2.2.1 Παράδειγμα μεταφοράς κόστους στον MBE-OPT-CONS(i)

Έχουμε αναφέρει ότι η μεταφορά κόστους στους MBE μπορεί να μας φέρει σαν αποτέλεσμα καλύτερο άνω φράγμα για τη συνάρτηση κόστους. Ας υλοποιήσουμε αυτή την τεχνική στο παράδειγμα που κάναμε για το MBE-OPT-CONS(i). Για λόγους ευκολίας θα αναφέρουμε το πρόβλημα πολύ συνοπτικά εδώ. Σε περίπτωση που δεν είμαστε αναλυτικοί για τα βήματα του MBE-OPT-CONS(i), καλείσθε να ανατρέξετε σε εκείνο το παράδειγμα και να καταλάβετε καλύτερα το τι συμβαίνει ακριβώς. Τέλος, πρέπει να πούμε ότι θα παραλείψουμε να αναφέρουμε το τι γίνεται με τους ισχυρούς περιορισμούς, καθώς δε συμμετέχουν στη μεταφορά κόστους, αλλά θα τους λαμβάνουμε σιωπηρά υπόψιν μας.

- ➔ $X = \{x_1,x_2,x_3,x_4,x_5,x_6\}$.
- ➔ $D = \{D_1,D_2,D_3,D_4,D_5,D_6\}$, με $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = \{0,1\}$.
- ➔ Ασθενείς περιορισμοί C_s:
  - $F_1(x_1)=\begin{Bmatrix}8,\text{εάν } x_1=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_2(x_2)=\begin{Bmatrix}6,\text{εάν } x_2=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_3(x_3)=\begin{Bmatrix}5,\text{εάν } x_3=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_4(x_4)=\begin{Bmatrix}2,\text{εάν } x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_5(x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_6(x_6)=\begin{Bmatrix}4,\text{εάν } x_6=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{12}(x_1,x_2)=\begin{Bmatrix}10,\text{εάν } x_1=x_2=1\\8,\text{εάν } x_1=x_2=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{13}(x_1,x_3)=\begin{Bmatrix}8,\text{εάν } x_1=x_3=1\\2,\text{εάν } x_1=1 \text{ και } x_3=0\\10,\text{εάν } x_1=0 \text{ και } x_3=0\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{235}(x_2,x_3,x_5)=\begin{Bmatrix}8,\text{εάν } x_2=x_3=x_5\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{124}(x_1,x_2,x_4)=\begin{Bmatrix}8,\text{εάν } x_1=x_2=x_4=1\\0,\text{αλλιώς}\end{Bmatrix}$
  - $F_{56}(x_5,x_6)=\begin{Bmatrix}10,\text{εάν } x_5=x_6=1\\6,\text{εάν } x_5=x_6=0\\0,\text{αλλιώς}\end{Bmatrix}$
- ➔ Ισχυροί περιορισμοί, $C_h = \{R_{12},R_{13},R_{14},R_{24},R_{25},R_{35}\}$, όπου $R_{12}=R_{13}=R_{14}=R_{24}=R_{25}=R_{35}=\{(1,0),(0,1),(0,0)\}$.

Θα χρησιμοποιήσουμε τη $d_2 = (x_6,x_1,x_5,x_2,x_3,x_4)$ και $i = 2$. Βάζουμε τους ισχυρούς και ασθενείς περιορισμούς σε κάδους.

Κάδος	Συναρτήσεις
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $R_{24}(x_2,x_4)$, $R_{14}(x_1,x_4)$
Bucket(x_3):	$F_3(x_3)$, $F_{235}(x_2,x_3,x_5)$, $F_{13}(x_1,x_3)$, $R_{35}(x_3,x_5)$, $R_{13}(x_1,x_3)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $R_{12}(x_1,x_2)$, $R_{25}(x_2,x_5)$
Bucket(x_5):	$F_5(x_5)$, $F_{56}(x_5,x_6)$
Bucket(x_1):	$F_1(x_1)$
Bucket(x_6):	$F_6(x_6)$

Πίνακας 16: Οι κάδοι μετά την αρχικοποίηση του MBE-OPT-CONS(i)

Στο Bucket(x_4), θα σχηματισθούν οι mini-bucket(x_1,x_4) και mini-bucket(x_2,x_4).

- Στο mini-bucket(x_1,x_4):
  - Έχουμε μόνο ισχυρούς περιορισμούς.
- Στο mini-bucket(x_2,x_4):
  - Έχουμε τις $F_4(x_4)$ και $F_{124}(x_1,x_2,x_4)$. Θα υπολογισθεί η $h^{x_4}(x_1,x_2)=\begin{Bmatrix}2,\text{εάν } x_2=0\,(x_4=1)\\0,\text{αλλιώς}\,(x_4=0)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_2).

Στο Bucket(x_3), θα σχηματισθούν οι mini-bucket(x_1,x_3) και mini-bucket(x_3,x_5).

Σε αυτόν τον κάδο, έχουμε δύο μικρο-κάδους που χωρίζουν τις συναρτήσεις του Bucket(x_3) σε δύο μέρη. Οι συναρτήσεις αυτές είναι οι $F_3(x_3)$, $F_{13}(x_1,x_3)$ και $F_{235}(x_2,x_3,x_5)$. Θα μεταφέρουμε κόστος από τη $F_{13}(x_1,x_3)$ στη $F_{235}(x_2,x_3,x_5)$. Η μεταφορά αυτή έχει δοθεί στο προηγούμενο παράδειγμα που είδαμε για μεταφορά κόστους. Θα μπορούσαμε να μεταφέρουμε και κόστος από τη $F_3(x_3)$ στη $F_{235}(x_2,x_3,x_5)$, αλλά δεν είναι αναγκαίο, καθώς και οι δύο συναρτήσεις βρίσκονται στον ίδιο μικρο-κάδο.

- Στο mini-bucket(x_1,x_3):
  - Έχουμε τη $F_{13}(x_1,x_3)$. Θα υπολογισθεί η $h^{x_3}(x_1)=\begin{Bmatrix}8,\text{εάν } x_1=0\,(x_3=0)\\2,\text{αλλιώς}\,(x_3=0)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_1).
- Στο mini-bucket(x_3,x_5):
  - Έχουμε τις $F_{235}(x_2,x_3,x_5)$ και $F_3(x_3)$. Θα υπολογισθεί η $h^{x_3}(x_2,x_5)=\begin{Bmatrix}2,\text{εάν } x_5=1\,(x_3=0)\\10,\text{εάν } x_5=x_2=0\,(x_3=0)\\5,\text{αλλιώς}\,(x_3=1)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_2).

Στο Bucket(x_2), θα σχηματισθούν οι mini-bucket(x_1,x_2) και mini-bucket(x_2,x_5).

Σε αυτόν τον κάδο έχουμε δύο μικρο-κάδους που χωρίζουν τις συναρτήσεις του Bucket(x_2) σε δύο μέρη. Οι συναρτήσεις αυτές είναι οι $F_2(x_2)$, $F_{12}(x_1,x_2)$, $h^{x_3}(x_2,x_5)$ και $h^{x_4}(x_1,x_2)$. Θα μεταφέρουμε κόστος από τις $F_2(x_2)$ και $h^{x_3}(x_2,x_5)$ στη $h^{x_4}(x_1,x_2)$.

- Στο mini-bucket(x_1,x_2):
  - Έχουμε $h^{x_4}(x_1,x_2)$ και $F_{12}(x_1,x_2)$. Θα υπολογισθεί η $h^{x_2}(x_1)=\begin{Bmatrix}12,\text{εάν } x_1=1\,(x_2=0)\\4,\text{αλλιώς}\,(x_2=0)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_1).
- Στο mini-bucket(x_2,x_5):
  - Έχουμε τις $h^{x_3}(x_2,x_5)$ και $F_2(x_2)$. Θα υπολογισθεί η $h^{x_2}(x_5)=\begin{Bmatrix}8,\text{εάν } x_5=0\,(x_2=1)\\0,\text{αλλιώς}\,(x_2=0)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_5).

Στο Bucket(x_5), θα σχηματισθεί ο mini-bucket(x_5,x_6).

- Στο mini-bucket(x_5,x_6):
  - Έχουμε τις συναρτήσεις $h^{x_2}(x_5)$, $F_5(x_5)$ και $F_{56}(x_5,x_6)$. Θα υπολογισθεί η

$h^{x_5}(x_6)=\begin{Bmatrix}14, \text{εάν } x_6=0\,(x_5=0)\\ 12, \text{αλλιώς}\,(x_5=1)\end{Bmatrix}$ και θα τοποθετηθεί στον Bucket(x_6).

Στο Bucket(x_1), θα σχηματισθεί ο mini-bucket(x_1).

- Στο mini-bucket(x_1):
  - Έχουμε τις $h^{x_3}(x_1)$, $h^{x_2}(x_1)$ και $F_1(x_1)$. Θα υπολογισθεί η $h^{x_1}=20\,(x_1=0)$ και θα τοποθετηθεί στον Bucket(x_6).

Στο Bucket(x_6), θα σχηματισθεί ο mini-bucket(x_6).

- Στο mini-bucket(x_6):
  - Έχουμε τις $h^{x_5}(x_6)$, h^{x_1} και $F_6(x_6)$. Θα υπολογισθεί η $h^{x_6}=36\,(x_6=1)$ και θα μας δώσει σαν άνω φράγμα το 36.

Καταπληκτικό, με τη μεταφορά κόστους βρήκαμε για άνω φράγμα 36, ενώ χωρίς τη μεταφορά είχαμε βρει 37. Είμαστε τυχεροί και το 36 είναι το βέλτιστο κόστος. Εντούτοις, κάτι τέτοιο συμβαίνει πολύ σπάνια. Δεν πρέπει να μπερδευτούμε και να υποθέσουμε ότι με τη μεταφορά κόστους βρίσκουμε πάντα τη βέλτιστη τιμή της συνάρτησης κόστους. Απλά, βελτιώνουμε το άνω φράγμα της.

Απομένει να βρούμε την αποτίμηση – λύση του προβλήματος και μέσω αυτής, να υπολογίσουμε το κάτω φράγμα.

Στο Bucket(x_6), πρέπει $x_6 = 1$.
Στο Bucket(x_1), πρέπει $x_1 = 0$.
Στο Bucket(x_5), πρέπει $x_5 = 1$.
Στο Bucket(x_2), πρέπει $x_2 = 0$.
Στο Bucket(x_3), πρέπει $x_3 = 0$.
Στο Bucket(x_4), πρέπει $x_4 = 1$.

Από τις αποτιμήσεις υπολογίζουμε το κάτω όριο:
$F_1(0) + F_2(1) + F_3(0) + F_4(0) + F_5(0) + F_6(0) + F_{12}(0,1) + F_{13}(0,0) + F_{235}(1,0,0) + F_{124}(0,1,0) + F_{56}(0,0) = 0 + 6 + 0 + 0 + 0 + 0 + 0 + 10 + 0 + 0 + 6 = 36$.

Όπως είπαμε, σχεδόν ποτέ ένας MBE με μεταφορά κόστους δε βρίσκει το βέλτιστο κόστος. Οι κάδοι στο τέλος:

Κάδος	Συναρτήσεις
Bucket(x_4):	$F_4(x_4)$, $F_{124}(x_1,x_2,x_4)$, $R_{24}(x_2,x_4)$, $R_{14}(x_1,x_4)$
Bucket(x_3):	$F_3(x_3)$, $F_{235}(x_2,x_3,x_5)$, $F_{13}(x_1,x_3)$, $R_{35}(x_3,x_5)$, $R_{13}(x_1,x_3)$
Bucket(x_2):	$F_2(x_2)$, $F_{12}(x_1,x_2)$, $R_{12}(x_1,x_2)$, $R_{25}(x_2,x_5)$, $h^{x_4}(x_1,x_2)$, $h^{x_3}(x_2,x_5)$
Bucket(x_5):	$F_5(x_5)$, $F_{56}(x_5,x_6)$, $h^{x_2}(x_5)$
Bucket(x_1):	$F_1(x_1)$, $h^{x_3}(x_1)$, $h^{x_2}(x_1)$
Bucket(x_6):	$F_6(x_6)$, $h^{x_5}(x_6)$, h^{x_1}

Πίνακας 17: Η τελική κατάσταση των κάδων μετά το πέρας της MBE-OPT-CONS(i)

7.2.2.2 Κριτήρια για βέλτιστη μεταφορά κόστους

Πολλές φορές, όταν έχουμε ένα σύνολο συναρτήσεων που μοιράζονται κάποιες μεταβλητές και θέλουμε να κάνουμε μεταφορά κόστους, δεν ξέρουμε από ποια συνάρτηση σε ποια είναι καλύτερα να γίνει η μεταφορά κόστους. Το σε ποια συνάρτηση θα γίνει η μεταφορά παίζει μεγάλο ρόλο για την ποιότητα του αποτελέσματος.

Είναι επιθυμητό το κόστος να μεταφερθεί σε μια συνάρτηση για την οποία, στο μικρο-κάδο που βρίσκεται, να υπάρχουν οι περισσότεροι ισχυροί περιορισμοί που να αναφέρονται στην εμβέλειά της. Το να μεταφερθεί το κόστος σε μια συνάρτηση που έχει τους λιγότερους ισχυρούς περιορισμούς που να αναφέρονται στην εμβέλειά της στον κάδο που ανήκει είναι το αντίθετο άκρο και μη επιθυμητό.

Θα κάνουμε δύο παρατηρήσεις με τις οποίες θα προσπαθήσουμε να οδηγήσουμε τη μεταφορά κόστους σε συνάρτηση ούτως ώστε να έχουμε όλο και πιο καλύτερη συνάρτηση κόστους.

Παρατήρηση 1

Έστω ότι έχουμε τις συναρτήσεις $F_{12}(x_1,x_2)$, $F_{23}(x_2,x_3)$, $F_{24}(x_2,x_4)$ και $F_{34}(x_3,x_4)$ και ότι εκτελούμε έναν MBE αλγόριθμο με i = 1 και d = (x_1,x_2,x_3,x_4). Χωρίζουμε τις συναρτήσεις σε τέσσερις κάδους.

Στον κάδο Bucket(x_4), υπάρχουν οι $F_{24}(x_2,x_4)$ και $F_{34}(x_3,x_4)$. Θα σχηματιστούν δύο μικρο-κάδοι, με τον κάθε κάδο να περιέχει από μια εκ των δύο συναρτήσεων. Αν θέλουμε να κάνουμε μεταφορά κόστους, από ποια συνάρτηση σε ποια θα την κάνουμε; Από τη $F_{24}(x_2,x_4)$ στη $F_{34}(x_3,x_4)$ ή αντίστροφα;

Απαλείφοντας τη x_4, θα δημιουργηθούν δύο συναρτήσεις, η $h^{x_4}(x_2,x_4)$ και η $h^{x_4}(x_3,x_4)$. Η $h^{x_4}(x_2,x_4)$ θα πάει στον κάδο Bucket(x_2) και η $h^{x_4}(x_3,x_4)$ θα πάει στον Bucket(x_3). Τον Bucket(x_3) θα τον επεξεργαστούμε πριν τον Bucket(x_2). Μας συμφέρει να μεταφέρουμε το κόστος στη $F_{34}(x_3,x_4)$, γιατί η $h^{x_4}(x_3,x_4)$ θα πάει σε κάδο που θα επεξεργαστεί νωρίς, και μετά όταν θα ξαναγίνει απαλοιφή, η νέα συνάρτηση θα πάει τελικά στο Bucket(x_2). Μας συμφέρει το μεταφερόμενο κόστος να επεξεργαστεί πολλές φορές. Όσο πιο πολλές φορές το φιλτράρουμε, τόσο καλύτερο θα είναι το τελικό αποτέλεσμα.

Κανόνας βέλτιστης μεταφοράς κόστους 1

> *Κάθε φορά που θα έχουμε να διαλέξουμε από που προς που θα μεταφερθεί το κόστος, κριτήριο μας για την απόφασή μας αυτή θα είναι το σε ποιον κάδο οδηγούν οι συναρτήσεις. Θα μεταφέρουμε το κόστος σε αυτήν που μας οδηγεί σε κάδο που απαλείφει μεταβλητή που είναι όσο πιο δεξιά γίνεται στην απαρίθμηση d.*

Παρατήρηση 2

Έστω ότι έχουμε τις τρεις συναρτήσεις $F_{7654}(x_7,x_6,x_5,x_4)$, $F_{7321}(x_7,x_3,x_2,x_1)$ και $F_{7651}(x_7,x_6,x_5,x_1)$ και έστω ότι είναι σε τρεις διαφορετικούς μικρο-κάδους. Έχουμε έξι διαφορετικούς τρόπους μεταφοράς κόστους. Ποιον θα επιλέξουμε; Παρατηρούμε ότι οι $F_{7654}(x_7,x_6,x_5,x_4)$ και $F_{7321}(x_7,x_3,x_2,x_1)$ έχουν στην εμβέλειά τους μία κοινή μεταβλητή, οι $F_{7321}(x_7,x_3,x_2,x_1)$ και $F_{7651}(x_7,x_6,x_5,x_1)$ δύο κοινές μεταβλητές και οι $F_{7654}(x_7,x_6,x_5,x_4)$ και $F_{7651}(x_7,x_6,x_5,x_1)$ τρεις. Το τελευταίο ζευγάρι είναι και το καλύτερο γιατί στη μεταφορά του κόστους συμμετέχουν οι πιο πολλές μεταβλητές, και πιθανώς η πιο πολλή πληροφορία. Οπότε θα επιλέξουμε μεταφορά κόστους μεταξύ $F_{7654}(x_7,x_6,x_5,x_4)$ και $F_{7651}(x_7,x_6,x_5,x_1)$, με οποιαδήποτε φορά, δεν έχουμε κάποια άλλη πληροφορία που να μας υποδεικνύει ποια φορά είναι η καλύτερη.

Κανόνας βέλτιστης μεταφοράς κόστους 2

> *Εάν θέλουμε να μεταφέρουμε κόστος μεταξύ συναρτήσεων και δεν ξέρουμε ποιες δύο να επιλέξουμε προτιμάμε αυτές που στην εμβέλειά τους έχουν τις πιο πολλές κοινές μεταβλητές.*

7.3 Βιβλιογραφικές αναφορές

Λίγο πιο αναλυτικά, μπορείτε να βρείτε τον DFS MBE στο [RELJ05]. Για ισοδύναμα δίκτυα με κόστη μπορείτε να δείτε το [RL06].

8 Επιλυτής Psarra

Ο επιλυτής Psarra είναι μια βιβλιοθήκη σε C++, που υλοποιεί απαλοιφή με κάδους (BE). Αρχικά, σχεδιάστηκε ώστε να επεκτείνει τον επιλυτή Naxos, ο οποίος είναι μια βιβλιοθήκη σε C++, η οποία επιλύει προβλήματα βελτιστοποίησης με περιορισμούς (COP), με τη βοήθεια μεθόδων αναζήτησης και του αλγορίθμου για συνέπεια ακμών AC-5.

8.1 Επιλυτής Naxos

Ο Naxos σχεδιάστηκε στα πλαίσια πτυχιακής εργασίας για το τμήμα πληροφορικής και τηλεπικοινωνιών του εθνικού και καποδιστριακού πανεπιστημίου Αθηνών, από το φοιτητή Νίκο Ποθητό, υπό την επίβλεψη του επίκουρου καθηγητή του τμήματος κ. Παναγιώτη Σταματόπουλο.

Ο Naxos σχεδιάστηκε σε αντικειμενοστραφές μοντέλο και υλοποιήθηκε σε C++ με σκοπό την επίλυση COP. Στο δίκτυο περιορισμών εφαρμόζεται ο αλγόριθμος AC-5 [DH91]. Έπειτα τρέχουμε αναζήτηση για το δίκτυο. Αρχικά, από το Naxos υλοποιήθηκε η πρώτα κατά βάθος αναζήτηση (DFS) και ουσιαστικά προσομοιώνεται η αναζήτηση με επέκταση και οριοθέτηση (BnB) με αλγόριθμο αναζήτησης τη DFS.

8.2 Επιλυτής Amorgos

Αργότερα, ο φοιτητής Φοίβος Θεοχάρης, του ίδιου τμήματος, πάλι υπό την επίβλεψη του κ. Σταματόπουλου, υλοποίησε για το Naxos και άλλες μεθόδους αναζήτησης, δημιουργώντας τη βιβλιοθήκη Amorgos. Συγκεκριμένα υλοποίησε τις:

- αναζήτηση πρώτα κατά βάθος
- αναζήτηση με περιορισμένη συμφωνία
- αναζήτηση διαμοιρασμού πίστωσης
- αναζήτηση φραγμένου βάθους με οπισθοδρόμηση
- αναζήτηση με περιορισμένες αναθέσεις
- αναζήτηση με φραγμένη κατά βάθος ασυμφωνία
- αναζήτηση με επαναληπτική διεύρυνση
- αναζήτηση με φραγμένη οπισθοδρόμηση
- αναζήτηση πρώτα κατά βάθος με επανεκκινήσεις
- αναζήτηση με σταδιακό περιορισμό πλάτους
- αναζήτηση με συναρτησιακό περιορισμό πλάτους
- επαναληπτική δειγματοληψία
- ένα-δείγμα

8.3 Επιλυτής Psarra

Ο επιλυτής Psarra πήρε το όνομά του, από το νησί του κεντρικού Αιγαίου, Ψαρά[11]. Η βάφτιση αυτή προέκυψε από το γεγονός ότι οι δύο προηγούμενες βιβλιοθήκες αναφέρονται και αυτές σε νησιά και επειδή τον ένα χρόνο που υπηρέτησα σαν καθηγητής δευτεροβάθμιας εκπαίδευσης στο νησί αυτό, συμπάθησα αρκετά τους μαθητές μου.

8.3.1 Στόχοι του επιλυτή Psarra

Στόχος του Psarra είναι η αντικειμενοστραφής υλοποίηση για την επίλυση COP με απαλοιφή με κάδους (BE) και απαλοιφή με μικρο-κάδους (MBE). Υπάρχει η προσπάθεια ο Psarra να ενσωματωθεί στο Naxos και να χρησιμοποιήσει τις ήδη υλοποιημένες κλάσεις, όπου είναι αυτό δυνατό. Επίσης θα υλοποιηθεί ο MBE σαν ευριστική συνάρτηση στη BnB. Στο τέλος της

11 Τα Ψαρρά τα τελευταία χρόνια γράφονται με ένα ρο – Ψαρά. Καθώς στην Ελληνική Επανάσταση η ορθογραφία τους είναι με δύο ρο, τιμητικά, κράτησα έστω στην greeklish αναπαράστασή τους αυτή την ορθογραφία.

υλοποίησής του, θα γίνει σύγκριση της απόδοσης του Psarra σε σχέση με αυτή του Naxos. Ουσιαστικά, θα γίνει σύγκριση του BE και της BnB για το αντικειμενοστραφές μοντέλο.

8.3.2 Η υλοποίηση του επιλυτή Psarra

Ο Psarra βρίσκεται ακόμα σε αρχικό στάδιο. Έως τώρα έχει υλοποιηθεί η επίλυση COP με BE. Δυστυχώς, ακριβώς επειδή ο Psarra βρίσκεται στο ξεκίνημα, υπάρχουν αρκετά μειονεκτήματα στην υλοποίηση του.

- Οι περιορισμοί σε αυτή τη νηπιακή φάση του Psarra αντιμετωπίζονται σαν πίνακες. Αυτό συνεπάγεται μεγάλη πολυπλοκότητα σε χώρο και χρόνο.
- Υπάρχουν κάποιες ιδέες για βελτιστοποίηση στους αλγορίθμους ELIM-OPT και MBE-OPT-CONS(i), που δεν έχουν ελεγχθεί, καθώς πρέπει πρώτα να υλοποιηθούν οι αρχικοί αλγόριθμοι όσο το δυνατόν καλύτερα, χωρίς τις δικές μας βελτιώσεις – ιδέες, καθώς αυτές επεμβαίνουν σε αυτούς τους αλγορίθμους.
- Έχει υλοποιηθεί ο αλγόριθμος ELIM-OPT, ενώ ο MBE-OPT-CONS(i) βρίσκεται υπό κατασκευή.

Οι κλάσεις που χρησιμοποιούνται είναι ουσιαστικά οι ίδιες με αυτές του Naxos. Η ουσιαστική διαφορά με το Naxos, είναι ότι υπάρχει διαχωρισμός των περιορισμών σε ισχυρούς και σε ασθενείς, κάτι που δεν υπάρχει διακριτά στο Naxos. Αυτή τη στιγμή, το σημείο στο οποίο βρίσκεται η πρόοδος του Psarra έγκειται στην απόφαση εάν θα πρέπει να υπάρχει αυτή η διακριτή διαφορά μεταξύ των περιορισμών, ή εάν θα πρέπει να ακολουθηθεί η τακτική του Naxos.

8.4 Μελλοντική εργασία

Η ανάπτυξη του Psarra θα συνεχιστεί και μετά τη διεκπεραίωση της διπλωματικής αυτής εργασίας. Στη λίστα με τις εργασίες που πρέπει να γίνουν, είναι τα:

- Αναπαράσταση περιορισμών σε κάτι καλύτερο από πίνακες, κάτι που γίνεται ήδη στο Naxos. Θα επικεντρωθούμε στην όσο καλύτερη μοντελοποίηση της απαλοιφής μεταβλητών στους περιορισμούς.
- Υλοποίηση εύρεσης και, όσο το δυνατόν, βέλτιστης απαρίθμησης για τον BE. Θα υλοποιηθούν αλγόριθμοι που υπάρχουν ήδη στο χώρο, ενώ θα υλοποιηθεί μία άλλη νέα ιδέα.
- Υλοποίηση απαλοιφής με μικρο-κάδους (MBE). Αφού υλοποιηθεί ο MBE, θα συνδυάσουμε MBE και BnB, όπως ακριβώς έχει αναλυθεί στο εδάφιο 6.3.
- Σύγκριση της απόδοσης μεταξύ του Naxos και του Psarra.
- Βελτίωση των ήδη υπαρχόντων αλγορίθμων για BE και MBE.

9 Βιβλιογραφία

[Rin03] Rina Dechter, Constraint Processing. Morgan Kaufmann, 2003

[RBT06] Francesca Rossi, Peter van Beek, Toby Walsh, Handbook of Constraint Programming. Elsevier, 2006

[Tsa93] Tsang, E.P.K., Foundations of Constraint Satisfaction. Academic Press, London and San Diego, 1993

[Rég94] Régin, J, A filtering algorithm for constraints of difference in CSPs. Proceedings of the 12th National Conference on AI, (vol. 1) 362-367, 1994

[Zha04] Yuanlin Zhang, On Tightness of Constraints.Principles and Practice of CP , 2004

[DH91] Y. Deville and P. Van Hentenryck, An efficient arc consistency algorithm for a class of CSP problems. Proceedings of the 12th IJCAI, 325-330, 1991

[MH86] Roger Mohr and Thomas C. Henderson, Arc and path consistency revisited. Artificial Intelligence, vol. 28, Issue 2, 225-233, 1986

[Coo89] Martin C. Cooper, An Optimal k-ConsistencyAlgorithm. Artificial Intelligence, vol. 41, Issue 1, 89-95, 1990

[SSW99] B. M. Smith, K. Stergiou, T. Walsh, Modelling the Golomb Ruler Problem.IJCAI-99 Workshop on Non Binary Constraints , 1999

[Sch99] T. Schiex, A note on CSP graph parameters.Technical report 1999/03, INRA , 1999

[Shc08] Shcherbina, O. A., Local elimination algorithms for solving sparse discrete problems. Vychislitelnoi Matematiki i Matematicheskoi Fiziki, vol. 48, Number 1, 152-167, 2008

[Lar01] J. Larrosa, On the time complexity of Bucket Elimination Algorithms.An ICS technical report , 2001

[KLR01] K. Kask, J. Larrosa, R. Dechter, Up and Down Mini-Buckets Scheme for Approximating COP.Technical Report, UCIrvine , 2001

[VLS96] Gerard Verfaillie, Michel Lematre, Thomas Schiex, Russian doll search for solving constraint optimization problems.Proceedings AAAI 1996 , 1996

[Rina99] R. Dechter, Bucket elimination: A unifying framework for reasoning.Artificial Intelligence , 1999

[KR94] Kalev Kask, Rina Dechter, A General Scheme for Automatic Generation of Search Heuristics. Artificial Intelligence, 129:91-131, 1994

[RR97] Dechter, R., and Rish, I., A scheme for approximating probabilistic inference."Uncertainty in Artificial Intelligence" (UAI97) , 1997

[Rin99] R. Dechter, Bucket Elimination: A Unifying Framework for Probabilistic Inference. Uncertainty in Artificial Intelligence, UA196, 211-219, 1996

[RKL01] R. Dechter, K. Kask, J. Larrosa, A General Scheme for Multiple Lower Bound Computation.Constraint Programming (CP2001) , 2001

[RKM02] Rina Dechter, Kalev Kask and Robert Mateescu, Iterative Join-Graph Propagation. Proceedings of Uncertainty in AI (UAI 2002), 128-136, 2002

[KRLC01] Kalev Kask, Rina Dechter, Javier Larrosa, Fabio Cozman, Unifying Tree-Decomposition Schemes for Automated Reasoning.Technical Report, UCIrvine , 2001

[LM03] Javier Larrosa, Enric Morancho, Solving 'Still Life' with Soft Constraints and Bucket Elimination. CP 2003, 466-479, 2003

[RELJ06] Rollón, Emma Larrosa, Javier, Bucket elimination for multiobjective optimization problems. Journal of Heuristics, Volume 12, 307-328(22), 2006

[RM08] Robert Mateescu and Rina Dechter, Mixed deterministic and probabilistic networks.ICS Technical Report , 2008

[RKBE02] Rina Dechter, Kalev Kask, Eyal Bin, Roy Emek, Generating Random Solutions for Constraint Satisfaction Problems.Proceedings of AAAI-2002 , 2002

[Zil96] Zilberstein S., Using anytime algorithms in intelligent systems. The AI magazine ISSN 0738-4602, vol. 17, Number 3, 73-83, 1996

[RELJ05] Rollón, Emma Larrosa, Javier, Depth-First Mini-Bucket Elimination. Lecture notes in computer science, vol. 3709, pp. 563-577, 2005

[RL06] Emma Rollon, Javier Larrosa, Mini-bucket Elimination with Bucket Propagation. CP 2006, 484-498, 2006

10 Γλωσσάρι / Αγγλικά σε Ελληνικά

adaptive consistency	προσαρμοζόμενη συνέπεια
approximation error	σφάλμα προσέγγισης
arc consistency	συνέπεια ακμών
arc-consistent edge	ακμή συνεπής ως προς τις ακμές
arc-consistent network	δίκτυο συνεπές ως προς τις ακμές
arc-consistent variable	μεταβλητή συνεπής ως προς τις ακμές
assignment	αποτίμηση
auction problem	πρόβλημα της δημοπρασίας
auctioneer	δημοπράτης
biset path consistent relative to a variable	δισύνολο μεταβλητών συνεπές ως προς το μονοπάτι σε σχέση με μια μεταβλητή
bounding evaluation function	συνάρτηση εκτίμησης οριοθέτησης
branch and bound search	αναζήτηση με επέκταση και οριοθέτηση
branch rearrangement	αναδιάταξη κλαδιών
bucket elimination algorithm	αλγόριθμος απαλοιφής με κάδους
children	παιδιά
complete algorithm	πλήρης αλγόριθμος
computation	υπολογισμός
computation tree	δέντρο υπολογισμού
consistent	συνεπές
constraint	περιορισμός
constraint biset relational path-consistent relative to x_i	δισύνολο περιορισμών σχεσιακά συνεπές ως προς τις ακμές σε σχέση με μια μεταβλητή x_i
constraint graph	γράφος με περιορισμούς
constraint network	δίκτυο με περιορισμοί
constraint optimization	βελτιστοποίηση με περιορισμούς
constraint problem	πρόβλημα με περιορισμούς
constraint processing	επεξεργασία περιορισμών
constraint relational arc-consistent relative to x_i	περιορισμός σχεσιακά συνεπής ως προς τις ακμές σε σχέση με μια μεταβλητή x_i
constraint satisfaction problem	πρόβλημα ικανοποίησης περιορισμών
constraint set relational m-consistent relative to x_i	σύνολο περιορισμών σχεσιακά συνεπές ως προς τις ακμές σε σχέση με μια μεταβλητή x_i
cost graph	γράφος με κόστη
cost network	δίκτυο με κόστη

criterion function	συνάρτηση κριτήριο
depth – first	πρώτα κατά βάθος
directional arc consistency	κατευθυνόμενη συνέπεια ακμών
directional i-consistency	κατευθυνόμενη i-συνέπεια
directional path consistency	κατευθυνόμενη συνέπεια μονοπατιού
directional relational m-consistency	κατευθυνόμενη σχεσιακή m-συνέπεια
dynamic programming	δυναμικός προγραμματισμός
equivalent cost networks	ισοδύναμα δίκτυα με κόστη
equivalent networks	ισοδύναμα δίκτυα
global cost function	συνολική συνάρτηση κόστους
globally consistent network	καθολικά συνεπές δίκτυο
graph width	πλάτος γράφου
hard constraint	ισχυρός περιορισμός
heuristic function	ευριστική συνάρτηση
horizontal compaction	οριζόντια σύμπτυξη
hyperedge	υπερακμή
hypergraph	υπεργράφος
i-consistency	i-συνέπεια
i-consistent network	i-συνεπές δίκτυο
inconsistent	ασυνεπής
induced graph	επαγόμενος γράφος
induced width of a graph	επαγόμενο πλάτος γράφου
induced width of an ordered graph	επαγόμενο πλάτος διατεταγμένου γράφου
inference	συμπερασμός
internal linear path	εσωτερικό γραμμικό μονοπάτι
mini-bucket elimination algorithm	αλγόριθμος απαλοιφής με μικρο-κάδους
minimal network	ελαχιστικό δίκτυο
n-ary constraint	ν-αδικός περιορισμός
network directional arc-consistent relative to order d	δίκτυο κατευθυνόμενα συνεπές ως προς τις ακμές σε σχέση με μια απαρίθμηση μεταβλητών d
network directional i-consistent relative to order d	δίκτυο κατευθυνόμενα i-συνεπές σε σχέση με μια απαρίθμηση μεταβλητών d
network directional path-consistent relative to order d	δίκτυο κατευθυνόμενα συνεπές ως προς το μονοπάτι σε σχέση με μια απαρίθμηση μεταβλητών d
network directionally relationally m-consistent	δίκτυο κατευθυνόμενα σχεσιακά m-συνεπές

network strongly directional i-consistent relative to order d	δίκτυο ισχυρά κατευθυνόμενα i-συνεπές σε σχέση με μια απαρίθμηση μεταβλητών d
node width	πλάτος κόμβου
objective function	αντικειμενική συνάρτηση
oracle	μαντείο
order	απαρίθμηση
ordered graph	διατεταγμένος γράφος
overlapping	επικαλυπτόμενος
parents	γονείς
partitioning	διαμέριση
path consistency	συνέπεια μονοπατιού
path consistent network	δίκτυο συνεπές ως προς το μονοπάτι
path consistent relation relative to a variable	σχέση συνεπής ως προς το μονοπάτι σε σχέση μια μεταβλητή
projection network	δίκτυο προβολής
Relational arc consistency	σχεσιακή συνέπεια ακμών
relational m-consistency	σχεσιακή m-συνέπεια
relational path consistency	σχεσιακή συνέπεια μονοπατιού
scope	εμβέλεια
search	αναζήτηση
search tree	δέντρο αναζήτησης
soft constraint	ασθενής περιορισμός
sound algorithm	έγκυρος αλγόριθμος
step	βήμα
strongly i-consistent network	ισχυρά i-συνεπές δίκτυο
tighter network than	πιο αυστηρό δίκτυο
transformation	μετασχηματισμός
tuple	πλειάδα
universal constraint	καθολικός περιορισμός
valid	έγκυρος
variable elimination algorithm	αλγόριθμος απαλοιφής μεταβλητών
vertical compaction	κάθετη σύμπτυξη
width of an ordering	πλάτος μιας απαρίθμησης

11 Γλωσσάρι / Ελληνικά σε Αγγλικά

i-συνέπεια	i-consistency
i-συνεπές δίκτυο	i-consistent network
ακμή συνεπής ως προς τις ακμές	arc-consistent edge
αλγόριθμος απαλοιφής με κάδους	bucket elimination algorithm
αλγόριθμος απαλοιφής με μικρο-κάδους	mini-bucket elimination algorithm
αλγόριθμος απαλοιφής μεταβλητών	variable elimination algorithm
αλγόριθμος πάσης στιγμής	anytime algorithm
αναδιάταξη κλαδιών	branch rearrangement
αναζήτηση	search
αναζήτηση με επέκταση και οριοθέτηση	branch and bound search
αντικειμενική συνάρτηση	objective function
απαρίθμηση	order
αποτίμηση	assignment
ασθενής περιορισμός	soft constraint
ασυνεπής	inconsistent
βελτιστοποίηση με περιορισμούς	constraint optimization
βήμα	step
γονείς	parents
γράφος με κόστη	cost graph
γράφος με περιορισμούς	constraint graph
δέντρο αναζήτησης	search tree
δέντρο υπολογισμού	computation tree
δημοπράτης	auctioneer
διαμέριση	partitioning
διατεταγμένος γράφος	ordered graph
δίκτυο ισχυρά κατευθυνόμενα i-συνεπές σε σχέση με μια απαρίθμηση μεταβλητών d	network strongly directional i-consistent relative to order d
δίκτυο κατευθυνόμενα i-συνεπές σε σχέση με μια απαρίθμηση μεταβλητών d	network directional i-consistent relative to order d
δίκτυο κατευθυνόμενα συνεπές ως προς τις ακμές σε σχέση με μια απαρίθμηση μεταβλητών d	network directional arc-consistent relative to order d
δίκτυο κατευθυνόμενα συνεπές ως προς το μονοπάτι σε σχέση με μια απαρίθμηση μεταβλητών d	network directional path-consistent relative to order d
δίκτυο κατευθυνόμενα σχεσιακά m-συνεπές	network directionally relationally m-consistent

δίκτυο με κόστη	cost network
δίκτυο με περιορισμοί	constraint network
δίκτυο προβολής	projection network
δίκτυο συνεπές ως προς τις ακμές	arc-consistent network
δίκτυο συνεπές ως προς το μονοπάτι	path consistent network
δισύνολο μεταβλητών συνεπές ως προς το μονοπάτι σε σχέση με μια μεταβλητή	biset path consistent relative to a variable
δισύνολο περιορισμών σχεσιακά συνεπές ως προς τις ακμές σε σχέση με μια μεταβλητή x_i	constraint biset relational path-consistent relative to x_i
δυναμικός προγραμματισμός	dynamic programming
έγκυρος	valid
έγκυρος αλγόριθμος	sound algorithm
ελαχιστικό δίκτυο	minimal network
εμβέλεια	scope
επαγόμενο πλάτος γράφου	induced width of a graph
επαγόμενο πλάτος διατεταγμένου γράφου	induced width of an ordered graph
επαγόμενος γράφος	induced graph
επεξεργασία περιορισμών	constraint processing
επικαλυπτόμενος	overlapping
εσωτερικό γραμμικό μονοπάτι	internal linear path
ευριστική συνάρτηση	heuristic function
ισοδύναμα δίκτυα	equivalent networks
ισοδύναμο δίκτυα με κόστη	equivalent cost networks
ισχυρά i-συνεπές δίκτυο	strongly i-consistent network
ισχυρός περιορισμός	hard constraint
κάθετη σύμπτυξη	vertical compaction
καθολικά συνεπές δίκτυο	globally consistent network
καθολικός περιορισμός	universal constraint
κατευθυνόμενη i-συνέπεια	directional i-consistency
κατευθυνόμενη συνέπεια ακμών	directional arc consistency
κατευθυνόμενη συνέπεια μονοπατιού	directional path consistency
κατευθυνόμενη σχεσιακή m-συνέπεια	directional relational m-consistency
μαντείο	oracle
μεταβλητή συνεπής ως προς τις ακμές	arc-consistent variable
μετασχηματισμός	transformation
ν-αδικός περιορισμός	n-ary constraint

οριζόντια σύμπτυξη	horizontal compaction
παιδιά	children
περιορισμός	constraint
περιορισμός σχεσιακά συνεπής ως προς τις ακμές σε σχέση με μια μεταβλητή x_i	constraint relational arc-consistent relative to x_i
πιο αυστηρό δίκτυο	tighter network than
πλάτος γράφου	graph width
πλάτος κόμβου	node width
πλάτος μιας απαρίθμησης	width of an ordering
πλειάδα	tuple
πλήρης αλγόριθμος	complete algorithm
πρόβλημα ικανοποίησης περιορισμών	constraint satisfaction problem
πρόβλημα με περιορισμούς	constraint problem
πρόβλημα της δημοπρασίας	auction problem
προσαρμοζόμενη συνέπεια	adaptive consistency
πρώτα κατά βάθος	depth – first
συμπερασμός	inference
συνάρτηση εκτίμησης οριοθέτησης	bounding evaluation function
συνάρτηση κριτήριο	criterion function
συνέπεια ακμών	arc consistency
συνέπεια μονοπατιού	path consistency
συνεπές	consistent
συνολική συνάρτηση κόστους	global cost function
σύνολο περιορισμών σχεσιακά συνεπές ως προς τις ακμές σε σχέση με μια μεταβλητή x_i	constraint set relational m-consistent relative to x_i
σφάλμα προσέγγισης	approximation error
σχέση συνεπής ως προς το μονοπάτι σε σχέση μια μεταβλητή	path consistent relation relative to a variable
σχεσιακή m-συνέπεια	relational m-consistency
σχεσιακή συνέπεια ακμών	Relational arc consistency
σχεσιακή συνέπεια μονοπατιού	relational path consistency
υπερακμή	hyperedge
υπεργράφος	hypergraph
υπολογισμός	computation

12 Ευρετήριο