Τεχνητή Νοημοσύνη

Ικανοποίηση Περιορισμών

Δρ. Δημήτριος Κουτσομητρόπουλος

Προβλήματα Ικανοποίησης Περιορισμών

- Είναι γνωστές μερικές ιδιότητες της τελικής κατάστασης
- Αναζητείται ένα στιγμιότυπο της τελικής κατάστασης
- Παραδείγματα προβλημάτων:
  - Χρονοπρογραμματισμός ενεργειών
  - Σχεδίαση ενεργειών παραγωγής
  - Διαχείριση πόρων

Αναπαράσταση Προβλήματος

- Ένα πρόβλημα ικανοποίησης περιορισμών (constraint satisfaction problem) ορίζεται από:
  - Ένα σύνολο *n* μεταβλητών $V_1, V_2, \ldots, V_n$,
  - Ένα σύνολο *n* πεδίων τιμών $D_1, \ldots D_n$, που αντιστοιχούν σε κάθε μεταβλητή έτσι ώστε $V_i \in D_i$, και
  - Ένα σύνολο σχέσεων (περιορισμών) $C_1, C_2, \ldots C_m$ όπου $C_i(V_k, \ldots, V_r)$ μια σχέση μεταξύ μεταβλητών του προβλήματος.
- Ανάλογα με το πόσες μεταβλητές περιλαμβάνει ένας περιορισμός χαρακτηρίζεται ως *μοναδιαίος* (*unary*), *δυαδικός* (*binary*) ή *ανώτερης* τάξης (*higher order*)

Λύση Προβλήματος Περιορισμών

- Λύση αποτελεί μια ανάθεση τιμών στις μεταβλητές του προβλήματος:

  $V_1 = d_1, V_2 = d_2, \ldots V_n = d_n$ (όπου $d_1 \in D_1$, $d_2 \in D_2, \ldots, d_n \in D_n$)

  τέτοια ώστε να ικανοποιούνται οι περιορισμοί $C_1, C_2, \ldots, C_m$
- Τα προβλήματα με D_i διακριτών τιμών αναφέρονται ως *προβλήματα ικανοποίησης περιορισμών*, ενώ αυτά με D_i συνεχών τιμών ως *προβλήματα επίλυσης περιορισμών*.
- Η αναζήτηση λύσης στα προβλήματα ικανοποίησης περιορισμών οδηγεί στο φαινόμενο της συνδυαστικής έκρηξης (combinatorial explosion). Γι’ αυτό απαιτούνται ειδικοί αλγόριθμοι για τη μείωση του χώρου αναζήτησης.

Παράδειγμα Προβλήματος (1)

- Ζητούμενο: Με ποια σειρά πρέπει να εισαχθούν τα προϊόντα Α, Β, Γ, Δ μέσα σε ένα βιομηχανικό μύλο.
- Απαιτήσεις: Το προϊόν Α πρέπει να εισαχθεί στο μύλο μετά από το Δ, το Γ πριν από το Β, και το Β πριν από το Α.
- Αναπαράσταση:
  - Μεταβλητές: V_A, V_B, V_Γ, V_Δ
  - Πεδία τιμών: $D_A \equiv D_B \equiv D_\Gamma \equiv D_\Delta = \{1, 2, 3, 4\}$
  - Περιορισμοί: $V_A \neq V_B \neq V_\Gamma \neq V_\Delta$ (διαφορετική σειρά)

    $V_A > V_\Delta$, $V_\Gamma < V_B$, $V_B < V_A$

Παράδειγμα Προβλήματος (2)

- Δυνατές λύσεις:
  - $V_A = 4$, $V_B = 2$, V_Γ, $= 1$, $V_\Delta = 3$ (Γ, Β, Δ, Α)
  - $V_A = 4$, $V_B = 3$, V_Γ, $= 1$, $V_\Delta = 2$ (Γ, Δ, Β, Α)
  - $V_A = 4$, $V_B = 3$, V_Γ, $= 2$, $V_\Delta = 1$ (Δ, Γ, Β, Α)

Παραγωγή και Δοκιμή(1)

- Χρήση Γεννήτριας Λύσεων (Generator) και Ελεγκτή (Tester)

- Αλγόριθμος
  - Παράγαγε μια υποψήφια λύση (συνδυασμό τιμών μεταβλητών) (γεννήτρια)
  - Έλεγξε αν είναι λύση (δηλ. ικανοποιεί τους περιορισμούς) (ελεγκτής)
  - Αν είναι σταμάτα (επιτυχία), αλλιώς πήγαινε στο βήμα 1.

Παραγωγή και Δοκιμή(2)

- Ιδιότητες/κριτήρια γεννήτριας λύσεων:

  - Να έχει πληρότητα (παράγονται όλοι οι δυνατοί συνδυασμοί τιμών-λύσεις)
  - Να είναι απέριττη (κάθε δυνατός συνδυασμός-λύση παράγεται μια φορά)
  - Να έχει ικανότητα ενημέρωσης (χρησιμοποιεί πληροφορία σχετική με το πρόβλημα για μείωση των παραγόμενων συνδυασμών-λύσεων)

Παραγωγή και Δοκιμή(3)

- Π.χ. στο πρόβλημα της σειράς εισαγωγής των προϊόντων στο βιομηχανικό μύλο, χρησιμοποιώντας ως πληροφορία σχετική με το πρόβλημα το ότι το προϊόν Α εισάγεται πάντα τελευταίο, παράγονται μόνο οι συνδυασμοί:

$V_A = 4$, $V_B = 1$, $V_Γ$, $= 1$, $V_Δ = 1$

$V_A = 4$, $V_B = 2$, $V_Γ$, $= 1$, $V_Δ = 1$

…

$V_A = 4$, $V_B = 4$, $V_Γ$, $= 4$, $V_Δ = 4$

Δηλ. 4^3 = <u>64</u> αντί 4^4 = <u>256</u>.

Αλγόριθμοι Επιδιόρθωσης (1)

- Για μείωση του χώρου αναζήτησης που δημιουργείται.
- Αναρρίχηση λόφου (Hill-climbing)

1. Ανάθεσε στις μεταβλητές τυχαίες τιμές από τα πεδία τιμών τους.
2. Αν οι τιμές των μεταβλητών δεν παραβιάζουν τους περιορισμούς του προβλήματος τότε επέστρεψε τις τιμές αυτές ως λύση.
3. Εξέτασε για κάθε μεταβλητή όλες τις δυνατές τιμές που μπορεί να πάρει.
   i. Αν κάποια από τις τιμές που εξετάστηκαν ελαχιστοποιεί το πλήθος των περιορισμών που παραβιάζονται, ανάθεσε την τιμή της στην αντίστοιχη μεταβλητή και επέστρεψε στο βήμα 2.
   ii. Αν δεν υπάρχει τιμή που να ελαχιστοποιεί το πλήθος των περιορισμών, τότε επέστρεψε στο βήμα 1 (τοπικό ελάχιστο–ο αλγόριθμος ξεκινά από μια νέα τυχαία ανάθεση τιμών).

Μειονεκτήματα: (α) Εξέταση μεγάλου πλήθους γειτονικών καταστάσεων

(β) Μπορεί να «πέσει» σε τοπικό ελάχιστο

Αλγόριθμοι Επιδιόρθωσης (2)

- Ευριστικός αλγόριθμος ελαχίστων συγκρούσεων

1. Ανέθεσε στις μεταβλητές τυχαίες τιμές από τα πεδία τιμών τους.
2. Αν οι τιμές των μεταβλητών δεν παραβιάζουν τους περιορισμούς του προβλήματος τότε επέστρεψε τις τιμές αυτές ως λύση.
3. Εξέτασε για μια τυχαία μεταβλητή όλες τις δυνατές τιμές που μπορεί να πάρει.
i. Αν κάποια από τις τιμές για τη μεταβλητή που εξετάστηκαν μειώνει το πλήθος των περιορισμών που παραβιάζονται, ανάθεσε την τιμή της στη μεταβλητή.
ii. Αν δεν υπάρχει τιμή που να μειώνει το πλήθος των περιορισμών που παραβιάζονται, τότε επέλεξε μια τιμή που να διατηρεί τον ίδιο αριθμό περιορισμών.
iii. Αν δεν υπάρχει ούτε τέτοια τιμή, τότε άφησε την τιμή της εξεταζόμενης μεταβλητής
4. Επέστρεψε στο βήμα 2.

Πλεονέκτημα: Εξέταση ΌΧΙ μεγάλου πλήθους γειτονικών καταστάσεων

Μειονέκτημα: Μπορεί να «πέσει» σε τοπικό ελάχιστο

Κλασικοί Αλγόριθμοι Αναζήτησης (1)

- Οι κλασικοί αλγόριθμοι αναζήτησης (π.χ. DFS, BFS) είναι δυνατό να χρησιμοποιηθούν και για την επίλυση των προβλημάτων ικανοποίησης περιορισμών.
- Αναπαράσταση
  - Μια *κατάσταση* αποτελείται από τις μεταβλητές του προβλήματος (με τις τιμές τους).
  - Υπάρχει ένας μόνο *τύπος τελεστή*, ο οποίος αντιστοιχεί στην ανάθεση τιμής σε μια *μη-δεσμευμένη μεταβλητή* (δηλ. μεταβλητή στην οποία δεν έχει ανατεθεί τιμή).
  - *Αρχική κατάσταση*: όλες οι μεταβλητές είναι μη-δεσμευμένες.
  - *Τελική κατάσταση*: ελέγχεται αν έχει γίνει ανάθεση τιμών σε όλες τις μεταβλητές, καθώς επίσης και αν ικανοποιούνται όλοι οι περιορισμοί του προβλήματος.

Κλασικοί Αλγόριθμοι Αναζήτησης (2)

- Καλύτερα αποτελέσματα έχουν οι ευριστικοί αλγόριθμοι αναζήτησης (π.χ. ο BestFS).
- Η ευριστική συνάρτηση αφορά την επιλογή της μεταβλητής για ανάθεση τιμής στο επόμενο βήμα. Στηρίζεται,
  - στην αρχή της συντομότερης αποτυχίας (επιλογή μεταβλητής με το μικρότερο πεδίο τιμών) και
  - στην αρχή της πιο περιορισμένης μεταβλητής (επιλογή της μεταβλητής που συμμετέχει στους περισσότερους περιορισμούς σε περίπτωση ισοδύναμων πεδίων τιμών).
- Μειονεκτήματα
  - Μη ικανοποιητική απόδοση για προβλήματα μεγάλου μεγέθους.
  - Δεν γίνεται ικανοποιητική εκμετάλλευση των περιορισμών (a posteriori έλεγχος).
  - Εξακολουθεί να υπάρχει το φαινόμενο της συνδυαστικής έκρηξης.

Αλγόριθμοι Ελέγχου Συνέπειας

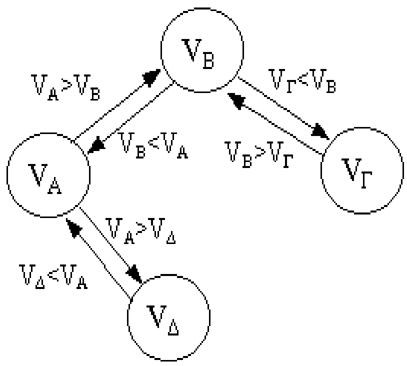
- Βασική ιδέα: απαλοιφή τιμών, που δεν είναι συνεπείς ως προς κάποιο περιορισμό, από τα πεδία τιμών (έλεγχος συνέπειας, που γίνεται a priori, δηλ. πριν την παραγωγή των πιθανών λύσεων, αντί για μετά-a posteriori).
- Γίνεται κατά κάποιο τρόπο διάδοση περιορισμών (constraint propagation), δηλ. οι μεταβολές σ' ένα πεδίο τιμών «διαδίδονται» στα πεδία των υπόλοιπων μεταβλητών μέσω των περιορισμών.

Τύποι Περιορισμών

- Ένας **μοναδιαίος περιορισμός** (unary constraint) αφορά μία μεταβλητή και ισοδυναμεί με καθορισμό του πεδίου τιμών.
- Ένας **δυαδικός περιορισμός** (binary constraint) συσχετίζει δύο μεταβλητές.
  - **Δυαδικό πρόβλημα ικανοποίησης περιορισμών** είναι ένα πρόβλημα που έχει μόνο μοναδιαίους και δυαδικούς περιορισμούς.
- Ένας **περιορισμός ανώτερης τάξης** (higher order constraint) συσχετίζει τρεις ή περισσότερες μεταβλητές.
  - π.χ., $Between(X,Y,Z)$, που μπορεί να οριστεί ως: $$\langle (X,Y,Z), X < Y < Z \textbf{ or } X > Y > Z \rangle$$
- Ένας περιορισμός που περιλαμβάνει αυθαίρετο αριθμό μεταβλητών ονομάζεται **καθολικός περιορισμός** (global constraint)
- Κάθε περιορισμός **με πεπερασμένα πεδία** μπορεί να αναχθεί σε ένα σύνολο δυαδικών περιορισμών, αν εισαχθούν αρκετές βοηθητικές μεταβλητές.

Γράφος Περιορισμών

- Ένα πρόβλημα μπορεί να αναπαρασταθεί ως γράφος (γράφος περιορισμών - constraint graph), όπου
  - ***τα τόξα (arcs)*** αναπαριστούν περιορισμούς
  - ***οι κόμβοι (nodes)*** αναπαριστούν τις μεταβλητές.



Περιορισμοί:

$V_A > V_\Delta$

$V_\Gamma < V_B$

$V_B < V_A$

Κατηγορίες Αλγορίθμων Συνέπειας

- **Αλγόριθμος συνέπειας κόμβου (Node Consistency)**
  - Μοναδιαίοι περιορισμοί (αφαιρεί τιμές πεδίων βασισμένος σε αυτούς).
- **Αλγόριθμοι συνέπειας τόξου (Arc Consistency-AC)**
  - Δυαδικοί περιορισμοί
  - Διάφοροι αλγόριθμοι συνέπειας τόξου, όπως οι AC-3, AC-4, AC-5, AC-6, κλπ.
  - Η δυσκολία που παρουσιάζουν οι αλγόριθμοι της κατηγορίας:
    - Διαγραφή μιας τιμής οδηγεί σε αλλαγές στα πεδία άλλων μεταβλητών.
    - Μετά από κάθε διαγραφή ασυνεπούς τιμής πρέπει να επανεξεταστούν τα πεδία των "άμεσα" συνδεδεμένων μεταβλητών.
- **Αλγόριθμοι συνέπειας μονοπατιού (path consistency algorithms)**
  - Περιορισμοί υψηλότερης τάξης
  - Υψηλό υπολογιστικό κόστος.

Συνέπεια Τόξου (Arc Consistency)

- Μια μεταβλητή V_i έχει συνέπεια τόξου ως προς μια άλλη μεταβλητή V_j, αν για κάθε τιμή στο τρέχον πεδίο D_i υπάρχει κάποια τιμή στο πεδίο D_j που ικανοποιεί τον δυαδικό περιορισμό στο τόξο (V_i, V_j).
- Παράδειγμα: Θεωρήστε τον περιορισμό $Y = X^2$, όπου το πεδίο τιμών τόσο της μεταβλητής X όσο και της Y είναι το σύνολο των δεκαδικών ψηφίων, $\{0,1, \dots, 9\}$.
  - Για να αποκτήσει η X συνέπεια τόξου ως προς την Y, μειώνουμε το πεδίο της X σε $\{0,1,2,3\}$.
  - Ομοίως, για να αποκτήσει η Y συνέπεια τόξου ως προς τη X, τότε το πεδίο της Y γίνεται $\{0,1,4,9\}$, και ολόκληρο το πρόβλημα ικανοποίησης περιορισμών αποκτά συνέπεια τόξου.
  - Δεν έχουμε λύσει ακόμη το πρόβλημα, αλλά περιορίσαμε πολύ τις δυνατές τιμές.

Συνέπεια Μονοπατιού (Path Consistency)

- Η **συνέπεια μονοπατιού** (path consistency) περιορίζει τους δυαδικούς περιορισμούς μέσω της χρήσης υπονοούμενων περιορισμών που προκύπτουν από την εξέταση τριάδων μεταβλητών.
  - Ένα σύνολο δύο μεταβλητών $\{V_i, V_j\}$ έχει **συνέπεια μονοπατιού** ως προς μια τρίτη μεταβλητή V_m αν, για κάθε ανάθεση τιμών $\{V_i = a, V_j = b\}$ συνεπή με τους περιορισμούς (αν υπάρχουν) ως προς τις μεταβλητές $\{V_i, V_j\}$, υπάρχει μια ανάθεση τιμής στην V_m που ικανοποιεί τους περιορισμούς ως προς τα σύνολα $\{V_i, V_m\}$ και $\{V_m, V_j\}$.
  - Η ονομασία αναφέρεται στη συνολική συνέπεια του μονοπατιού από τη X_i έως τη X_j με τη X_m ενδιάμεσα.

Κ-ΣΥΝΕΠΕΙΑ

Ένα πρόβλημα ικανοποίησης περιορισμών έχει ***k*-συνέπεια** αν, για οποιοδήποτε σύνολο k−1 μεταβλητών και για οποιαδήποτε συνεπή ανάθεση τιμών σε αυτές τις μεταβλητές, μια συνεπής τιμή μπορεί πάντα να ανατεθεί σε οποιαδήποτε k-οστή μεταβλητή.

Ένα πρόβλημα ικανοποίησης περιορισμών έχει **ισχυρή *k*-συνέπεια** (strong k-consistency) αν έχει k-συνέπεια και έχει επίσης $(k-1)$-συνέπεια, $(k-2)$-συνέπεια,… μέχρι και 1-συνέπεια.

- Ο αλγόριθμος συνέπειας κόμβου εξασφαλίζει ότι ο γράφος είναι ισχυρά 1-συνεπής.
- Οι αλγόριθμοι συνέπειας τόξου εξασφαλίζουν ισχυρή 2-συνεπεία.
- Προφανώς σε ένα γράφο με n κόμβους, εάν εξασφαλισθεί ότι ο γράφος είναι ισχυρά n-συνεπής, τότε
  - Μπορεί να βρεθεί λύση χωρίς αναζήτηση.
  - Ο συνολικός χρόνος εκτέλεσης είναι μόνο $O(n^2 d)$, όπου d το πλήθος των τιμών στο πεδίο κάθε μεταβλητής.
  - Όμως σε προβλήματα με k>2 το υπολογιστικό κόστος εφαρμογής είναι υψηλό, οπότε προτιμάται ο συνδυασμός το πολύ 2-συνέπειας με κλασσικό αλγόριθμο αναζήτησης.

Ο αλγόριθμος AC-3

- Ο απλούστερος αλγόριθμος συνέπειας τόξου.
- Έστω οι μεταβλητές V_1, V_2,..V_n με τιμές d_1, d_2,…,d_n από τα πεδία τιμών των μεταβλητών D_1, D_2,…,D_n και ένα σύνολο περιορισμών $C(V_i,V_j)$ για τις μεταβλητές αυτές, οι οποίοι αναπαριστώνται ως τόξα (V_i,V_j). Για συντομία, κάθε τόξο (V_i,V_j) αναφέρεται ως (i,j).
- Το Q περιλαμβάνει αρχικά όλα τα τόξα του γράφου περιορισμών.

Επανέλαβε τα ακόλουθα βήματα μέχρι το Q να γίνει κενό:

1. **Επέλεξε ένα τόξο (i,j) και διέγραψε το από το Q**
2. **Για κάθε τιμή d_i του πεδίου της μεταβλητής V_i έλεγξε αν υπάρχει τουλάχιστον μία τιμή d_j του πεδίου της μεταβλητής V_j τέτοια ώστε να ικανοποιεί το περιορισμό $C(V_i,V_j)$ που αντιστοιχεί στο τοξο (i, j).**
3. **Αν δεν υπάρχει τέτοια τιμή d_j τότε αφαίρεσε την τιμή d_i από το πεδίο τιμών της V_i. Αν το πεδίο τιμών της V_i είναι κενό τότε τερμάτισε με αποτυχία.**
4. **Αν έχει μεταβληθεί το πεδίο τιμών της V_i τότε πρόσθεσε στο σύνολο Q όλα τα τόξα (k,i), που αντιστοιχούν στους περιορισμούς $C(V_k,V_i)$, για $k \neq i$.**

Παράδειγμα

- Πρόβλημα: Σειρά εισαγωγής των προϊόντων A, B, Γ, Δ σ' ένα βιομηχανικό μύλο.

Περιορισμοί:

$V_A \neq V_B$ **(C1)**	$V_B \neq V_\Gamma$ **(C4)**	$V_A > V_\Delta$ **(C7)**
$V_A \neq V_\Gamma$ **(C2)**	$V_B \neq V_\Delta$ **(C5)**	$V_\Gamma < V_B$ **(C8)**
$V_A \neq V_\Delta$ **(C3)**	$V_\Gamma \neq V_\Delta$ **(C6)**	$V_B < V_A$ **(C9)**

Τα πεδία τιμών των μεταβλητών:

$V_A \in \{1,2,3,4\}$

$V_B \in \{1,2,3,4\}$

$V_\Gamma \in \{1,2,3,4\}$

$V_\Delta \in \{1,2,3,4\}$

ΕΠΑΝΕΞΕΤΑΣΗ

Λόγω C9 ($V_B < V_A$):	Λόγω C8 ($V_\Gamma < V_B$):	Λόγω C7 ($V_A > V_\Delta$):	Λόγω C9 ($V_B < V_A$):
$V_A \in \{\cancel{1},2,3,4\}$	$V_A \in \{2,3,4\}$	$V_A \in \{2,3,4\}$	$V_A \in \{\cancel{2},3,4\}$
$V_B \in \{1,2,3,\cancel{4}\}$	$V_B \in \{\cancel{1},2,3\}$	$V_B \in \{2,3\}$	$V_B \in \{2,3\}$
$V_\Gamma \in \{1,2,3,4\}$	$V_\Gamma \in \{1,2,\cancel{3},\cancel{4}\}$	$V_\Gamma \in \{1,2\}$	$V_\Gamma \in \{1,2\}$
$V_\Delta \in \{1,2,3,4\}$	$V_\Delta \in \{1,2,3,4\}$	$V_\Delta \in \{1,2,3,\cancel{4}\}$	$V_\Delta \in \{1,2,3\}$

Χαρακτηριστικά AC-3

- Προϋποθέτει δυαδικούς περιορισμούς.
  - Απαιτείται μετασχηματισμός περιορισμών ανώτερης τάξεως σε πρόβλημα δυαδικών περιορισμών (binarization).
- Μη-πληρότητα (υπάρχουν τιμές στα πεδία που δεν εμφανίζονται στη λύση)
  - Στο προηγούμενο παράδειγμα στο πεδίο τιμών της μεταβλητής V_A παρέμεινε η τιμή 3 (που δεν είναι δυνατόν να συμμετέχει στη λύση):

$V_A \in \{3,4\}$
$V_B \in \{2,3\}$
$V_\Gamma \in \{1,2\}$
$V_\Delta \in \{1,2,3\}$

- Οι αλγόριθμοι συνέπειας τόξου δεν απαλείφουν *όλες* τις ασυνεπείς τιμές.
- Οπότε για την επίλυση προβλημάτων περιορισμών χρησιμοποιούνται συνήθως αλγόριθμοι ελέγχου συνέπειας τόξου σε συνδυασμό με κάποιο κλασσικό αλγόριθμο αναζήτησης (DFS, BFS, BestFS).
- Πολυπλοκότητα; $\boldsymbol{O}(\boldsymbol{cd}^3)$, *c πλήθος τόξων, d πλήθος τιμών*
  - Έλεγχος ενός τόξου: *O* (d^2)
  - Επανεξέταση: κάθε τόξο (V_m, V_i) μπορεί να ελεγχθεί το πολύ *d* φορές.

Συνδυασμός Αλγορίθμων Συνέπειας και Κλασικής Αναζήτησης (1)

- Ο συνδυασμός αλγορίθμων συνέπειας και αναζήτησης στηρίζεται στη συμπληρωματικότητά τους:
  - Αλγόριθμοι συνέπειας: μη-πλήρεις αλλά αποδοτικοί
  - Κλασικοί αλγόριθμοι αναζήτησης: πλήρεις αλλά μη-αποδοτικοί
- Βασική ιδέα :

Μείωση του χώρου αναζήτησης με την χρήση ενός αλγορίθμου συνέπειας πριν από κάθε βήμα ανάθεσης τιμών (a priori pruning).

Υπάρχουν **τρεις βασικοί τρόποι συνδυασμού**, που

- έχουν κοινό το ότι εφαρμόζεται ένας αλγόριθμος συνέπειας πριν την εκκίνηση της διαδικασίας αναζήτησης,
- διαφέρουν στο βαθμό ελέγχου των πεδίων των μεταβλητών σε κάθε βήμα, δηλ. στον τρόπο διάδοσης των περιορισμών.

Συνδυασμός Αλγορίθμων Συνέπειας και Κλασικής Αναζήτησης (2)

- ***Ο προοπτικός έλεγχος (Forward checking)***
  - Απαλείφει τιμές από τα πεδία των μη-δεσμευμένων μεταβλητών που συνδέονται άμεσα με περιορισμούς με την μεταβλητή στην οποία μόλις ανατέθηκε τιμή.
    - Παραμένει μεγάλος αριθμός ασυνεπών τιμών στα πεδία.
    - Χαμηλό υπολογιστικό κόστος κάθε βήματος.
- ***Ο αλγόριθμος έγκαιρης μερικής εξέτασης (Partial Look Ahead)***
  - Κατευθυντική συνέπεια (directional consistency) σε κάθε βήμα (εξετάζει όλα τα πεδία τιμών των μη-δεσμευμένων μεταβλητών με προκαθορισμένη σειρά, ελέγχοντας τους περιορισμούς μία μόνο φορά).
    - Παραμένουν στα πεδία των μεταβλητών μη συνεπείς τιμές.
- ***Ο αλγόριθμος έγκαιρης πλήρους εξέτασης (Full Look Ahead) ή διατήρησης συνέπειας τόξου (Maintaining Arc Consistency - MAC).***
  - Εφαρμόζει πλήρη αλγόριθμο συνέπειας τόξου σε κάθε βήμα.
    - Αφαιρεί το μεγαλύτερο αριθμό ασυνεπών τιμών από τους τρεις.
    - Υψηλό υπολογιστικό κόστος

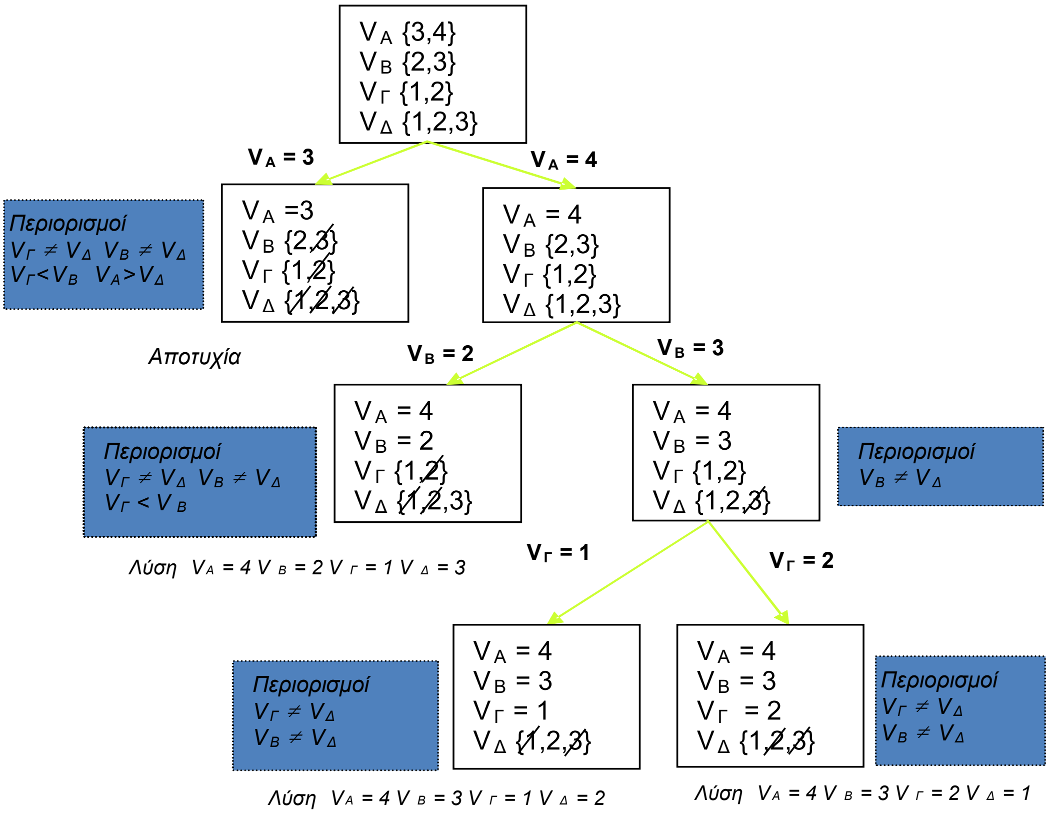
Συνδυασμός Αλγορίθμων Συνέπειας και Κλασικής Αναζήτησης (3)

Ο ολοκληρωμένος αλγόριθμος διατήρησης συνέπειας τόξου :

1. Για κάθε περιορισμό αφαίρεσε από τα πεδία τιμών των μεταβλητών τις τιμές εκείνες που δεν μπορούν να συμμετέχουν στην τελική λύση με ένα αλγόριθμο ελέγχου συνέπειας.
2. Στο μειωμένο χώρο αναζήτησης που προκύπτει από το προηγούμενο βήμα εφάρμοσε έναν κλασικό αλγόριθμο αναζήτησης για να βρεθεί η λύση.

   Σε κάθε βήμα (ανάθεση τιμής) αυτής της αναζήτησης εφάρμοσε ξανά τον αλγόριθμο ελέγχου συνέπειας έτσι ώστε να αφαιρεθούν τυχόν τιμές από τα πεδία των μεταβλητών οι οποίες δεν μπορούν να συμμετέχουν στην λύση.

Παράδειγμα



Πρόβλημα 8 Βασιλισσών(1)

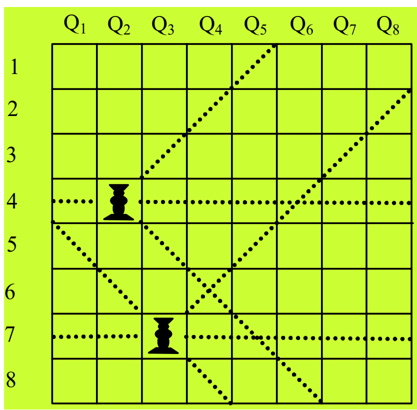
- Κλασικό παράδειγμα προβλήματος περιορισμών.
  - Το πρόβλημα απαιτεί να τοποθετηθούν 8 βασίλισσες σε μια σκακιέρα 8x8 χωρίς να απειλούν η μια την άλλη.
  - Το πρόβλημα ορίζεται και για περισσότερες των 8 βασιλισσών
  - Η δυσκολία στην επίλυσή του αυξάνει εκθετικά.
  - Χρησιμοποιείται για την μέτρηση της απόδοσης αλγορίθμων ικανοποίησης περιορισμών.
- Συνθήκη μη απειλής μεταξύ των βασιλισσών:

  Όλες οι βασίλισσες πρέπει να είναι σε διαφορετική γραμμή:

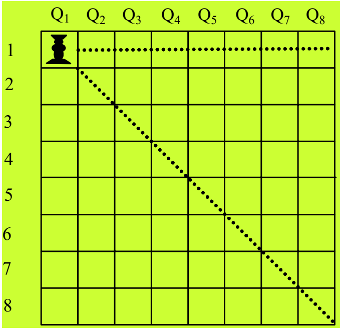
  $\forall$ i, j: Qj ≠ Qi.
  - Ισχύουν οι περιορισμοί:

    Qj ≠ Qj+n + n, για n>1 και n+j ≤8

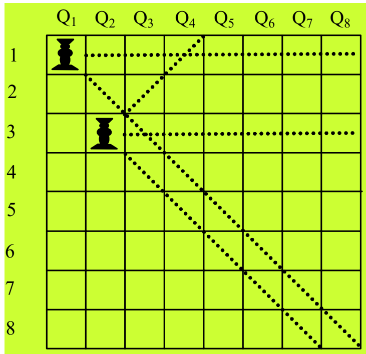
    Qj ≠ Qj+n – n, για n>1 και n+j ≤8
  - Σχηματική αναπαράσταση περιορισμών με δύο βασίλισσες στην σκακιέρα.



Πρόβλημα 8 Βασιλισσών (2)

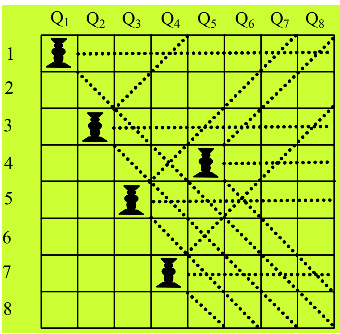


1. Ανάθεση τιμής στην πρώτη βασίλισσα (Q1=1)
2. Αφαίρεση τιμών που δεν ικανοποιούν τους περιορισμούς (βλ. σχήμα) από τα πεδία των μεταβλητών των υπολοίπων βασιλισσών

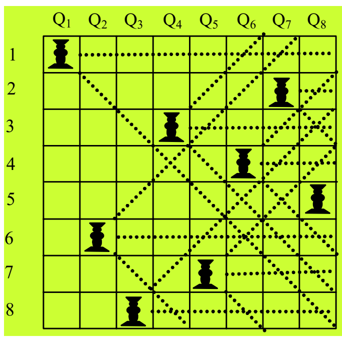


3. Ανάθεση τιμής στη δεύτερη βασίλισσα (Q2=3)
4. Αφαίρεση τιμών που δεν ικανοποιούν τους περιορισμούς (βλ. σχήμα) από τα πεδία των μεταβλητών των υπολοίπων βασιλισσών

Πρόβλημα 8 Βασιλισσών (3)



Περαιτέρω ανάθεση τιμών που δεν οδηγεί σε λύση.



Λύση στο πρόβλημα των 8 βασιλισσών