

ΥΣ02 Τεχνητή Νοημοσύνη - Χειμερινό Εξάμηνο 2025-2026

Εργασία 3 (2 μονάδες του συνολικού βαθμού στο μάθημα)

Ημερομηνία Ανακοίνωσης: 13/11/2025

Ημερομηνία Παράδοσης: 19/12/2025, ώρα 23:59

Αντιγραφή: Σε περίπτωση που προκύψουν φαινόμενα αντιγραφής, οι εμπλεκόμενοι θα βαθμολογηθούν **στην άσκηση με βαθμό μηδέν**.

Χρήση του ChatGPT: Επιτρέπεται χωρίς κανένα περιορισμό η χρήση του ChatGPT σε όσες ερωτήσεις θέλετε. Σε όποιες περιπτώσεις το χρησιμοποιήσετε όμως, πρέπει να δώσετε όλη την αλληλεπίδραση σας με το σύστημα μαζί με την απάντησή σας. **Προσοχή:** Η απάντησή σας πρέπει να γραφτεί πλήρως από εσάς και να ακολουθεί όσα μάθατε από τις διαλέξεις και τις διαφάνειες και δεν πρέπει να είναι “αντιγραφή” από την έξοδο του ChatGPT. Το ChatGPT μπορεί να σας βοηθήσει πάρα πολύ, αλλά στις περισσότερες περιπτώσεις δεν θα σας δώσει την απάντηση όπως την περιμένουμε με βάση τα όσα έχετε διδαχθεί.

Πρόβλημα 1: (The radio link frequency assignment problem - RLFA)

Στο ερώτημα αυτό έχετε να λύσετε το πρόβλημα ικανοποίησης περιορισμών RLFA (radio link frequency assignment) χρησιμοποιώντας τους αλγόριθμους FC, MAC και FC-CBJ. Το πρόβλημα RLFA ορίζεται ως εξής. Έχουμε να αναθέσουμε συχνότητες σε ραδιοφωνικές συνδέσεις. Οι μεταβλητές του προβλήματος είναι οι ραδιοφωνικές συνδέσεις και τα πεδία των μεταβλητών είναι οι διαθέσιμες συχνότητες. Τα πεδία είναι σύνολα θετικών αριθμών. Δίδονται επίσης δυαδικοί περιορισμοί ανάμεσα στις μεταβλητές. Οι περιορισμοί είναι της μορφής $|x - y| > k$ όπου x και y είναι μεταβλητές και k ένας θετικός αριθμός. Θέλουμε να βρούμε μια ανάθεση συχνοτήτων (τιμών) στις ραδιοφωνικές αναθέσεις (μεταβλητές) η οποία ικανοποιεί όλους τους περιορισμούς. Περισσότερες λεπτομέρειες για το πρόβλημα μπορείτε να βρείτε στην ιστοσελίδα <https://miat.inrae.fr/schiex/rlfap.shtml>.

Έχετε να κάνετε τα εξής:

1. Δείτε το συμπίεσμένο φάκελο `rlfap.zip` που έχουμε δημοσιεύσει μαζί με την άσκηση. Ο φάκελος αυτός περιέχει 12 στιγμιότυπα (instances) του προβλήματος για να δοκιμάσετε τους αλγόριθμους σας. Διαβάστε το αρχείο `odigies.txt` για να καταλάβετε το περιεχόμενο του φακέλου.

2. Υλοποιήστε τους αλγόριθμους FC, MAC και FC-CBJ και εφαρμόστε τους στο πρόβλημα. Προτείνεται να χρησιμοποιήσετε dynamic variable ordering με χρήση της ευρετικής dom/wdeg από το άρθρο
F. Boussemart, F. Hemery, C. Lecoutre and L. Sais. *Boosting Systematic Search by Weighting Constraints*. Proc. of ECAI 2004, pages 146–150, 2004. Διαθέσιμο στην ιστοσελίδα
<http://www.frontiersinai.com/ecai/ecai2004/ecai04/pdf/p0146.pdf> .
Θα χρειαστεί επίσης να γράψετε κώδικα για να διαβάζετε τα παραπάνω αρχεία, και να τα φορτώνετε σε όποιες δομές χρησιμοποιείτε στη μνήμη για τις μεταβλητές, τα πεδία και τους περιορισμούς.
Η υλοποίησή σας πρέπει να βασιστεί πάνω στον κώδικα Python που διατίθεται στη σελίδα <https://github.com/aimacode/aima-python/blob/master/csp.py> .
3. Να συγκρίνετε πειραματικά τους αλγόριθμους που υλοποιήσατε χρησιμοποιώντας τα στιγμιότυπα που σας δώσαμε και ορίζοντας κατάλληλες μετρικές. Θα πρέπει να εξηγήσετε ποια κριτήρια σύγκρισης χρησιμοποιήσατε και γιατί. Να παρουσιάσετε τα αποτελέσματά σας με ευκρίνεια χρησιμοποιώντας πίνακες και να τα σχολιάσετε.
4. Να δοκιμάσετε να λύσετε τα δοσμένα στιγμιότυπα με τον αλγόριθμο τοπικής αναζήτησης MINCONFLICTS. Να σχολιάσετε τα αποτελέσματα που προκύπτουν και να τα συγκρίνετε με αυτά του προηγούμενου ερωτήματος.

(100 μονάδες)

Πρόβλημα 2:

Πολλά προβλήματα ικανοποίησης περιορισμών είναι ταυτόχρονα και προβλήματα βελτιστοποίησης. Δηλαδή, κάθε εφικτή λύση, με βάση τους περιορισμούς του προβλήματος, έχει και ένα κόστος που προκύπτει από την εφαρμογή μίας αντικειμενικής συνάρτησης (objective function) επάνω στη λύση. Το ζητούμενο είναι να βρεθεί εκείνη η λύση με το ελάχιστο κόστος.

Θεωρήστε το εξής πρόβλημα: Μία αεροπορική εταιρεία έχει προγραμματίσει να εκτελέσει για μία προκαθορισμένη χρονική περίοδο N πτήσεις, στις οποίες μπορεί να αναφέρεται κανείς με τους κωδικούς $1, 2, 3, \dots, N$. Επιπλέον, μέσω κάποιας μεθόδου που εφάρμοσε, έχει δημιουργήσει M συνδυασμούς πτήσεων (pairings) P_i ($1 \leq i \leq M$). Δηλαδή, κάθε P_i περιλαμβάνει κάποιες από τις πτήσεις $1, 2, 3, \dots, N$ και, επίσης, τα P_i δεν είναι κατ' ανάγκη ξένα μεταξύ τους. Οι συνδυασμοί αυτοί είναι έτσι κατασκευασμένοι ώστε να είναι δυνατόν λόγω κανονισμών, συμβάσεων κλπ. να πραγματοποιηθούν ο καθένας, δηλαδή όλες οι πτήσεις που περιλαμβάνει, με έναν από τους διαθέσιμους κυβερνήτες της εταιρείας. Το ζητούμενο είναι να επιλεγούν κάποιοι συνδυασμοί που καλύπτουν ακριβώς τις πτήσεις της εταιρείας, με σκοπό να ανατεθούν σε συγκεκριμένους κυβερνήτες. Δηλαδή, δεν πρέπει ούτε να μείνει πτήση χωρίς κυβερνήτη, ούτε κάποια πτήση να έχει δύο ή περισσότερους κυβερνήτες. Τέλος, αν ένας συνδυασμός πτήσεων P_i έχει ένα κόστος (έξοδα διανυκτερεύσεων, αποζημιώσεις εκτός έδρας, πληρωμή υπερωριών κλπ.) στην εταιρεία C_i , οι συνδυασμοί που θα επιλεγούν πρέπει να είναι αυτοί που προκαλούν το

ελάχιστο συνολικό κόστος. Για να γίνει περισσότερο σαφές το ζητούμενο, δείτε το εξής παράδειγμα: Έστω $N = 5$, $M = 7$ και

i	P_i	C_i
1	{1, 2}	6
2	{1, 2, 3}	7
3	{1, 5}	5
4	{2}	3
5	{3, 4}	4
6	{4, 5}	8
7	{5}	6

Τότε, η βέλτιστη λύση είναι η $\{\{1, 5\}, \{2\}, \{3, 4\}\}$, με κόστος 12 ($=5+3+4$), ενώ οι άλλες πιθανές λύσεις $\{\{1, 2\}, \{3, 4\}, \{5\}\}$ και $\{\{1, 2, 3\}, \{4, 5\}\}$ έχουν κόστη 16 ($=6+4+6$) και 15 ($=7+8$), αντίστοιχα.

1. Μοντελοποιήστε το πρόβλημα αυτό σαν πρόβλημα ικανοποίησης περιορισμών (μεταβλητές, πεδία, περιορισμοί) και διατυπώστε την αντικειμενική συνάρτηση κόστους βάσει της οποίας θα πρέπει να βρεθεί η βέλτιστη λύση. Δώστε την απάντησή σας με ένα ξεκάθαρο μαθηματικό φορμαλισμό.
2. Υλοποιήστε σε οποιαδήποτε γλώσσα προγραμματισμού από τις συνήθειες (Python, C++, Java, κλπ.) την επίλυση του προβλήματος, επάνω στα datasets του συμπιεσμένου φακέλου www.di.uoa.gr/~takis/pairings.zip. Ο φάκελος αυτός περιέχει 10 στιγμιότυπα (instances) του προβλήματος για να δοκιμάσετε τη λύση σας. Διαβάστε το αρχείο format.txt για να καταλάβετε τη δομή των στιγμιότυπων NNxMMM.txt. Μπορείτε είτε να υλοποιήσετε αλγορίθμους που συζητήθηκαν στο μάθημα και είναι κατάλληλοι για την αντιμετώπιση του προβλήματος, είτε να εκμεταλλευτείτε υπάρχουσες βιβλιοθήκες προγραμματισμού με περιορισμούς, όπως:
 - Gecode (C++ library): <https://www.gecode.dev/>
 - Naxos solver (C++ library): <https://github.com/pothitos/naxos>
 - Choco (Java library): <https://choco-solver.org/>
 - python-constraint (Python library): <https://pypi.org/project/python-constraint/>
 - ECLiPSe (Prolog with constraint libraries): <https://www.eclipseclp.org/>
3. Θα πρέπει η υλοποίησή σας να βρίσκει τη βέλτιστη λύση, καθώς και το κόστος της, για κάθε στιγμιότυπο.
4. Για το πρόβλημα αυτό, θα πρέπει να παραδώσετε, εκτός από τον κώδικά σας, και μία αναλυτική αναφορά για τη δουλειά σας.

(100 μονάδες)

Πρόβλημα 3:

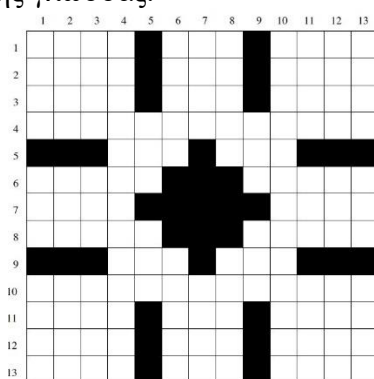
Η Αλίκη (Α), ο Βασίλης (Β), ο Γιώργος (Γ), η Δήμητρα (Δ), η Ελένη (Ε) και ο Ζαχαρίας (Ζ) έχουν καθίσει γύρω από ένα κυκλικό τραπέζι έξι θέσεων. Η Ελένη κάθεται δύο θέσεις δεξιότερα του Γιώργου, η Δήμητρα δύο θέσεις δεξιότερα του Ζαχαρία και ο Βασίλης τρεις θέσεις δεξιότερα της Ελένης. Το ζητούμενο είναι να βρεθεί η ακριβής (σχετική) θέση των ατόμων στο τραπέζι.

1. Μοντελοποιήστε το πρόβλημα αυτό σαν πρόβλημα ικανοποίησης περιορισμών, ορίζοντας τις απαιτούμενες μεταβλητές, τα πεδία τους και τους μεταξύ τους περιορισμούς. Τους περιορισμούς μπορείτε να τους διατυπώσετε είτε με μαθηματικό τρόπο, είτε μέσω συνόλου επιτρεπτών πλειάδων (όχι κατ' ανάγκη όλους με τον ίδιο τρόπο). Φροντίστε η μοντελοποίησή σας να είναι τέτοια ώστε να αποκλείει την εύρεση συμμετρικών λύσεων.
2. Εφαρμόστε πλήρη συνέπεια ακμών στον γράφο περιορισμών και δείξτε βήμα-βήμα πώς τροποποιούνται τα πεδία των μεταβλητών.
3. Βρείτε όλες τις λύσεις του προβλήματος χρησιμοποιώντας BT και εφαρμόζοντας συνέπεια επιπέδου FC μετά από κάθε βήμα ανάθεσης. Χρησιμοποιήστε το ευρετικό MRV για την επιλογή μεταβλητής, σε συνδυασμό με το ευρετικό βαθμού (degree heuristic) για την επίλυση ισοπαλιών. Όσον αφορά την επιλογή τιμής για μια μεταβλητή, χρησιμοποιήστε τη σειρά των τιμών όπως τις περιλάβατε στα πεδία του ερωτήματος 1.

(10+10+10=30 μονάδες)

Πρόβλημα 4:

Έστω ότι έχετε ένα κενό σταυρόλεξο, όπως αυτό του σχήματος, και ένα πολύ μεγάλο λεξικό που περιλαμβάνει λέξεις της Ελληνικής γλώσσας.



Το πρόβλημα που έχετε να αντιμετωπίσετε είναι να γεμίσετε το σταυρόλεξο (οριζόντια και κάθετα) με λέξεις από το λεξικό (τουλάχιστον δύο γραμμάτων).

1. Προτείνετε δύο **ουσιωδώς διαφορετικές** μοντελοποιήσεις του προβλήματος αυτού ως πρόβλημα ικανοποίησης περιορισμών. Δηλαδή, εξηγήστε ποιες μεταβλητές έχετε σε

κάθε μοντελοποίηση, τα πεδία τους και τους περιορισμούς τους.

2. Για κάθε μία από τις δύο μοντελοποιήσεις που προτείνετε, θεωρήστε ότι θα εφαρμόσετε τη μέθοδο της οπισθοδρόμησης (BT) με πρώτο έλεγχο (FC) για να επιλύσετε το πρόβλημα. Πώς θα συγκεκριμενοποιούσατε, για κάθε μοντελοποίηση, το ευρετικό επιλογής μεταβλητής MRV και πώς το ευρετικό επιλογής τιμής LCV;

(15+15=30 μονάδες)

Πρόβλημα 5:

Θεωρήστε τη μοντελοποίηση του προβλήματος των 6 βασιλισσών με τον κλασικό τρόπο, δηλαδή με 6 μεταβλητές $X_1, X_2, X_3, X_4, X_5, X_6$ που αντιστοιχούν στις γραμμές του πλαισίου 6×6 , με πεδίο τιμών στην καθεμία το σύνολο των ακεραίων $\{1, 2, 3, 4, 5, 6\}$, όπου η ανάθεση $X_i = j$ σημαίνει ότι στην i γραμμή τοποθετήθηκε βασίλισσα στην j στήλη. Οι περιορισμοί είναι οι γνωστοί, δηλαδή δεν επιτρέπεται να υπάρχουν δύο βασίλισσες στην ίδια γραμμή (εξασφαλισμένος από τη μοντελοποίηση), ίδια στήλη ή ίδια διαγώνιο. Έστω ότι επιλύουμε το πρόβλημα με τη μέθοδο της οπισθοδρόμησης (BT) σε συνδυασμό με την εφαρμογή κάποιας τεχνικής συνέπειας (FC ή MAC). Η σειρά αποτίμησης των μεταβλητών είναι λεξικογραφική (πρώτα η X_1 , μετά η X_2 , κοκ.) και σε κάθε μεταβλητή ανατίθεται η ελάχιστη τιμή που είναι διαθέσιμη στο πεδίο της.

Έχοντας ξεκινήσει με την ανάθεση $X_1 = 1$, μπορούμε να δούμε εύκολα ότι τελικά θα καταλήξουμε σε αδιέξοδο και ότι αυτή η μερική ανάθεση δεν μπορεί να επεκταθεί σε πλήρη εφικτή λύση. Οπότε, την αναιρούμε και συνεχίζουμε με την ανάθεση $X_1 = 2$.

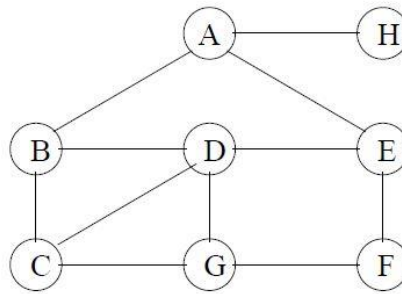
1. Αν μετά από κάθε βήμα ανάθεσης της BT εφαρμόζουμε και FC, θα οδηγηθούμε σε εφικτή λύση χωρίς να χρειαστεί να κάνουμε άλλη οπισθοδρόμηση; Αν ναι, σε ποια λύση; Αν όχι, σε ποια ανάθεση από αυτές που θα γίνουν στη συνέχεια θα χρειαστεί να οπισθοδρομήσουμε; Αιτιολογήστε την απάντησή σας δείχνοντας μετά από κάθε βήμα ανάθεσης τα πεδία των μεταβλητών που δεν έχουν πάρει ακόμα τιμή.
2. Ίδιο ερώτημα με το προηγούμενο, αλλά αντί για FC, να εφαρμόζουμε MAC μετά από κάθε ανάθεση.

(10+20=30 μονάδες)

Πρόβλημα 6:

Όταν σε ένα πρόβλημα ικανοποίησης περιορισμών έχουμε εφαρμόσει j -συνέπεια για κάθε $j = 1, 2, \dots, k$, λέμε ότι το πρόβλημα είναι **ισχυρά k -συνεπές**.

Θεωρήστε τον παρακάτω γράφο περιορισμών ενός προβλήματος ικανοποίησης περιορισμών.



1. Εξηγήστε γιατί αν εφαρμόσουμε ισχυρή 8-συνέπεια στο πρόβλημα αυτό, είναι δυνατόν να βρούμε λύση του σε χρόνο $O(nd)$, όπου n το πλήθος των μεταβλητών του και d το μέγιστο μέγεθος των πεδίων τους.
2. Όμως, επειδή η επιβολή ισχυρής 8-συνέπειας κοστίζει πολύ, επιλέγουμε να εφαρμόσουμε μόνο ισχυρή 3-συνέπεια στο πρόβλημα. Δείξτε ότι και πάλι, είναι δυνατόν, για το συγκεκριμένο πρόβλημα, να βρούμε λύση του σε χρόνο $O(nd)$.
3. Εξηγήστε πώς θα έπρεπε να εργασθούμε ώστε για το οποιοδήποτε πρόβλημα ικανοποίησης περιορισμών να βρούμε το ελάχιστο k , τέτοιο ώστε αν εφαρμόσουμε ισχυρή k -συνέπεια στο πρόβλημα, να μπορούμε να βρούμε λύση του σε χρόνο $O(nd)$.

(5+10+15=30 μονάδες)