Εργασία 3η - Τεχνητή Νοημοσύνη 2020

Στοιχεία Φοιτητή:
Ονοματεπώνυμο: Λουκοβίτης Γεώργιος
ΑΜ: 1115201800100

constrainSatisfaction.py

Dom/Wdeg heuristic:
Η δοθείσα συνάρτηση backtracking_search τροποποιείται ώστε να ενημερώνει ένα dictionary βαρών για κάθε περιορισμό σε περίπτωση αποτυχίας μιας ανάθεσης. Τα βάρη αυτή συνυπολογίζονται για διάταξη των μεταβλητών προς ανάθεση

FC Algorithm:
Χρησιμοποιείται η συνάρτηση backtrack του δοσμένου csp.py με τον inference ορισμένο στην συνάρτηση forward_checking. Έτσι ύστερα από την ανάθεση τιμής σε μία μεταβλητή γίνεται η διάδοση περιορισμών που προβλέπεται απο τον αλγόριθμο FC.

MAC Algorithm:
Χρησιμοποιείται η συνάρτηση backtrack του δοσμένου csp.py με τον inference ορισμένο στην συνάρτηση mac. Έτσι ύστερα από την ανάθεση τιμής σε μία μεταβλητή γίνεται η διάδοση περιορισμών που προβλέπεται απο τον αλγόριθμο MAC.

FC-CBJ Algorithm:
Αρχικά υλοποιείται ο αλγόριθμος CBJ. Η υλοποίηση του αλγορίθμου CBJ προκύπτει από την τροποποίηση το BT ώστε να διατηρεί ένα conflict set για κάθε μεταβλητή στο οποίο προστίθενται οι μεταβλητες, οι οποίες κατά την ανάθεση τους, μέσω του inference, αφαίρεσαν κάποια τιμή από την προαναφερθείσα μεταβλητή (η πληροφορία αυτή εξάγεται από την λίστα removals που επιστρέφει το inference). Η πληροφορία αυτή στην πορεία συνδυάζεται με ένα ιστορικό αναθέσεων, ώστε, σε περίπτωση αποτυχίας, ο αλγόριθμος να υπαναχωρήσει στην πιο πρόσφατα ανατεθειμένη μεταβλητή που στέρησε κάποια τιμή από την μεταβλητή που απέτυχε.
Με την χρήση του δοσμένου forward_checking για το inference στον παραπάνω αλγόριθμο υλοποιείται η συνάρτηση FC-CBJ

Min Conflict Algorithm:
Χρησιμοποιείται ο δοσμένος αλγόριθμος MinConflict με τον μέγιστο αριθμό βημάτων περιορισμένο στα 1000, για λόγους χρόνου, και στις μεταβλητές ανακατεμένες σε κάθε εκτέλεση με την (φρούδα) ελπίδα επιτυχίας

main:
Το πρόβλημα προς επίλυση προσδιορίζεται με το αναγνωριστικό του από μια καθολική μεταβλητή στην αρχή του κώδικα. Τα αρχεία με παραμέτρων του προβλήματος μετατρέπονται στις αναγκαίες δομές μέσω βοηθητικών συναρτήσεων

Σημείωση:
Για την εύρεση των αρχείων προβλημάτων αυτά πρέπει να βρίσκονται σε ένα φάκελο με το όνομα rlfap ο οποίος θα βρίσκεται στον ίδιο φακέλο με το constrainSatisfaction.py

Dependencies:
-numpy
-sortedcontainers

Εκτέλεση:
~ python3 constrainSatisfaction.py

Μετρήσεις:
Ως κριτήρια σύγκρισης των αλγορίθμων επιλέχθηκαν η επιτυχία στην εύρεση λύσης, ο αριθμός αναθέσεων που πραγματοποιήθηκαν και ο χρόνος που απαιτήθηκε. Ο αριθμός αναθέσεων επιλέγεται για να φανεί η αποτελεσματικότητα στην πρόβλεψη μελλοντικών dead ends και την αποτροπή τους, ενώ ο χρόνος επιλέγεται για να ληφθεί υπόψη το overhead που επιβάλλεται από πιο πολύπλοκους αλγορίθμους
Οι μετρήσεις (οσες πρόλαβαν να πραγματοποιηθούν σε κάποιο εύλογο χρονικό διάστημα και δεν οδήγησαν σε recursion error) περιέχονται σε ξεχωριστό αρχείο. Φαίνεται πως ο αλγόριθμος mac χρειάζεται τον μικρότερο αριθμό αναθέσεων για λύσει ένα πρόβλημα ή να συμπεράνει πως δεν μπορεί να λυθεί, ωστόσο το overhead που προσθέτει η διατήρηση του arc consistency τον οδηγεί σε τόσο μεγαλύτερους χρόνους απο τους υπόλοιπους αλγορίθμους που καθίσταται απαγορευτικός για μεγάλα προβλήματα. Ύστερα, οι αλγόριθμοι fc και fc-cbj κινούνται αρκετά κοντά μεταξύ τους, κάτι αναμενόμενο δεδομένου του ότι όσα τμήματα του δέντρου αναζήτησης

κλαδεύονται απο το bj κλαδεύονται και απο τον fc και ο cbj αποτελεί επέκταση του bj. Όσες διακυμάνσεις οφείλονται στην τυχαιότητα που εισάγεται από συναρτήσεις όπως η argminrand. Τέλος ο min conflict καταλήγει ο πιο αναξιόπιστος όλων των αλγορίθμων καθώς για την επιτυχία απαιτουνται εξαιρετικά παρά πολλές εκτελέσεις με τις μεταβλητές να ανακατεύονται πριν κάθε μία από αυτές, όπου και πάλι σε προβλήματα όπως το 8-f10 μετά από 2000!!!! εκτελέσεις απέτυχε να παράξει λύση