

Readme

main.py

Η συνάρτηση readFiles επιστρέφει τις μεταβλητές , τα πεδία ορισμού και τους γείτονες .

Επειτα αυτά και η συνάρτηση constraints που είναι στο αρχείο ai3 , δίνονται ως ορισμα για την κατασκευή του προβλήματος. Επειτα αυτό δίνεται σαν ορισμα στον αλγόριθμο backtrack με σκοπο να βρούμε την λύση στο πρόβλημα μας.

Fileread.py

Το αρχείο αυτό περιέχει την συνάρτηση readFiles, η οποία , διαβάζει τα τρία αρχεία και δημιουργεί και συμπληρώνει σωστά τις εκάστοτε δομές δεδομένων , όπως ζητούνται από το CSP.py

Πάνω πάνω στο name βάζουμε την κατάληξη των αρχείων που θέλουμε να τρέξουμε

ai3.py

Κανουμε το ίδιο για το name όπως στην fileread . Η συνάρτηση constraints μας βολεύει να είναι σε αυτό το αρχείο για να βρίσκουμε ευκολα τις φορές που κληθηκε.

Για κάθε περιορισμο $C(\alpha, \beta)$, δημιουργουμε και τον αντιστροφο , $C(\beta, \alpha)$

Οι αλλαγές που καναμε στις ήδη ετοιμες συναρτησεις AC3 , FC είναι απλα να δινουμε σαν ορισμα και να επιστρεφουμε το weight. Ακομη , στο σημειο που οι συναρτησεις θα επιστρεψουν false επειδη ενα domain αδειάζει τοτε , αυξανουμε τοσο το $weight[var, neigh]$ οσο και το $weight[neigh, var]$

Παρομοιες αλλαγές καναμε και στην backtrack , της οποιας προσθεσαμε ορισμενους μετρητες οι οποιοι μας βοηθουν στη εξαγωγή συμπερασμάτων

Η ευρετική dom/wdeg , επιστρεφει το $\min(||domain(var)|| / wdeg(var))$, [var not in assignment]

Results comments

Γενικά σχόλια.

Δεν έχουμε υλοποιήσει τον αλγόριθμο FC-OBJ . Ακόμη κάναμε αρκετές προσπάθειες με τον minConflicts , ωστόσο τα αποτελέσματα ήταν απογοητευτικά επομένως δεν τα έχουμε στις μετρήσεις μας. Αυτό οφείλεται στο γεγονός ότι ο minConflicts είναι κακίστος αλγόριθμος για αυτού του είδους τα προβλήματα

Επομένως τα δεδομένα μας για μερικά από τα δοσμένα στιγμιότυπα είναι:

```
2-f24
===FC===
average time = 0.21674521764119467
average num of nodes expanded 463
average num of constraints checked 38156
===mac===
average time = 0.5732386112213135
average num of nodes expanded 303
average num of constraints checked 185203
```

```
11
===FC===
average time = 13.000217517217001
average num of nodes expanded 6228
average num of constraints checked 1423018
===mac===
average time = 24.238382895787556
average num of nodes expanded 2837
average num of constraints checked 6223568
```

```
2-f25

===FC===
average time = 115.53299236297607
average num of nodes expanded 135709
average num of constraints checked 25328659
===mac===
average time = 238.78838543891908
average num of nodes expanded 30513
average num of constraints checked 62783956
```

6-w2

```
===FC===  
average time = 0.1493070920308431  
average num of nodes expanded 253  
average num of constraints checked 47198  
===mac===  
average time = 0.2694255510965983  
average num of nodes expanded 42  
average num of constraints checked 100243
```

Γενικά παρατηρούμε ότι ορισμένες φορές ο FC τρέχει πιο γρήγορα από τον MAC, και ελέγχει λιγότερες φορές τους περιορισμούς. Ωστόσο ο mac κάνει πολύ λιγότερες assignment

Ωστόσο έχουμε παραδείγματα όπως τα παρακάτω όπου ο FC αργεί περισσότερο από τον mac.

```
7-w1-f4  
===FC===  
average time = 6.743340492248535  
average num of nodes expanded 31094  
average num of constraints checked 1169395  
===mac===  
average time = 0.8799160321553549  
average num of nodes expanded 479  
average num of constraints checked 348772
```

Πιο συγκεκριμένα στο παρακάτω παράδειγμα, ο FC αργεί υπερβολικά πολύ για να τον μετρήσουμε, ενώ ο mac βγάζει εν τέλει λύση:

```
7-w1-f5  
===mac===  
average time = 70.84050333499908  
average num of nodes expanded 8828  
average num of constraints checked 34278487
```