ΠΑΝΟΣ ΛΕΛΑΚΗΣ

1083712

4ο ΕΤΟΣ

Μελέτη και υλοποίηση του παζλ Eternity II ως πρόβλημα MILP

Αυτή η εργασία αφιερώνεται στη μνήμη του Άλεξ, που αγαπούσε τα μαθηματικά.

Περίληψη

Το θέμα της εργασίας είναι η μαθηματική μοντελοποίηση και υλοποίηση του παζλ Eternity II (EII) ως πρόβλημα βελτιστοποίησης Μεικτού Ακεραίου και Γραμμικού Προγραμματισμού (MILP). Το EII είναι ένα παζλ 16x16 τετραγώνων που απαιτεί την τοποθέτηση 256 κομματιών έτσι ώστε οι πλευρές τους να ταιριάζουν με τις γειτονικές. Το πρόβλημα θεωρείται εξαιρετικά δύσκολο και υπολογιστικά βαρύ, καθώς από όταν διατέθηκε στην αγορά δεν έχει βρεθεί ολοκληρωμένη λύση από τους μαθηματικούς και προγραμματιστές που το έχουν μελετήσει.

Στην εργασία παρουσιάζεται το μαθηματικό μοντέλο του προβλήματος, το οποίο περιλαμβάνει τις μεταβλητές απόφασης, την αντικειμενική συνάρτηση και τους περιορισμούς που απαιτούνται για τη σωστή τοποθέτηση και περιστροφή των κομματιών. Αναπτύχθηκε επίσης κώδικας σε Python για την υλοποίηση του μοντέλου και τη χρήση του CBC solver για τον υπολογισμό της βέλτιστης λύσης, όπως επίσης και για την γραφική απεικόνιση της λύσης και την επεξεργασία των δεδομένων της. Το μοντέλο επεκτείνεται σε παρόμοια (Eternity-like) παζλ, επιτρέποντας τον ορισμό διαφορετικών μεγεθών ταμπλό και συνδυασμών σχημάτων και χρωμάτων.

Η εργασία καταλήγει με την παρουσίαση των αποτελεσμάτων για διάφορα Eternity-like παζλ, μέσω παραδειγμάτων χρήσης του προγράμματος, καθώς και με την λήψη συμπερασμάτων που αφορούν την επίδραση των διαφόρων μεταβλητών των παζλ στην επίδοση της εφαρμογής.

Ο κώδικας του προγράμματος, όπως επίσης και όλα τα αρχεία που το συνοδεύουν, είναι διαθέσιμα στο link: https://github.com/PanosLelakis/EII-MILP-solver [1].

I. ΕΙΣΑΓΩΓΗ

Το Eternity II (EII) είναι ένα παζλ σκέψης που δημιουργήθηκε από τον μαθηματικό Christopher Monckton. Από την στιγμή που διατέθηκε στην αγορά το καλοκαίρι του 2007, ανακοινώθηκε διαγωνισμός με έπαθλο 2 εκατομμυρίων δολαρίων για τον πρώτο που θα έλυνε το παζλ. Μέχρι την ημερομηνία λήξης του διαγωνισμού, που ήταν η τελευταία μέρα του 2010, δεν βρέθηκε ολοκληρωμένη λύση.

Το EII αποτελείται από ένα ταμπλό 16×16, σύνολο 256, θέσεων, στο οποίο ο παίκτης καλείται να τοποθετήσει 256 τετράγωνα κομμάτια, έτσι ώστε τα χρώματα και τα σχέδια της κάθε πλευράς κάθε κομματιού να συμφωνούν με αυτά των γειτονικών πλευρών σε όλη την επιφάνεια του ταμπλό. Επίσης, οι γκρι άκρες πρέπει να βρίσκονται στην περίμετρο του ταμπλό, σχηματίζοντας ένα εξωτερικό πλαίσιο.

Παρότι απλό στην λογική του, κανείς από τους χιλιάδες μαθηματικούς και προγραμματιστές που έχουν ασχοληθεί με το παζλ δεν έχει αναπτύξει αποδοτικό αλγόριθμο που να βρίσκει την ολοκληρωμένη λύση σε λογικό χρονικό διάστημα. Ο λόγος πίσω από την αποτυχία αυτή έγκειται στο γεγονός ότι ο Monckton σχεδίασε το παζλ έτσι ώστε να είναι ιδιαίτερα δύσκολο να λυθεί με το χέρι και υπολογιστικά βαρύ να λυθεί από υπολογιστή. Μάλιστα, ο ίδιος ανέφερε σε συνέντευξή του ότι «ακόμα και ο ισχυρότερος υπολογιστής του κόσμου δεν θα κατάφερνε να βρει την λύση πριν το τέλος του κόσμου».

Για βοήθεια, μαζί με το παζλ δίνεται ένα στοιχείο (hint) για την λύση του, δηλαδή αποκαλύπτεται η θέση και ο προσανατολισμός ενός από τα 256 κομμάτια πάνω στο ταμπλό. Εκτός από αυτά, μαζί με το αρχικό παζλ διατέθηκαν στην αγορά 4 clue puzzles, τα οποία έχουν διαστάσεις 6×6 και 12×6. Η λύση τους, η οποία υπολογίζεται γρήγορα και εύκολα από υπολογιστή, δίνει 4 επιπλέον hints για την λύση του EII.

Η απουσία ολοκληρωμένης λύσης για το EII εγείρει το ενδιαφέρον της κοινότητάς του και παρακινεί το κοινό να το μελετήσει μαθηματικά, όπως κάνει και η παρούσα εργασία. Συγκεκριμένα, όπως ζητήθηκε από την εκφώνηση, περιγράφεται το μαθηματικό μοντέλο που υλοποιεί το πρόβλημα ως πρόβλημα βελτιστοποίησης μεικτού ακεραίου και γραμμικού προγραμματισμού (Mixed Integer Linear Programming - MILP). Για την ολοκληρωμένη περιγραφή του, εξηγείται η λογική των μεταβλητών απόφασής του, της αντικειμενικής συνάρτησής του και των περιορισμών του. Για τον σκοπό της εργασίας, αναπτύχθηκε κώδικας σε Python που αρχικά μεταφράζει το μαθηματικό μοντέλο MILP σε κώδικα και χρησιμοποιεί τον CBC solver για την εύρεση της βέλτιστης λύσης του. Η λύση, ύστερα, απεικονίζεται γραφικά με τα κομμάτια ενωμένα πάνω στο ταμπλό και παρέχονται στον χρήστη χρήσιμες πληροφορίες σχετικά με αυτή.

Η εργασία δεν μελετά μόνο το original 16×16 EII, αλλά επεκτείνει το μοντέλο ώστε να περιγράφει κάθε Eternity-like παζλ. Πιο συγκεκριμένα, το μέγεθος του ταμπλό, το πλήθος των δυνατών συνδυασμών χρωμάτων και σχεδίων κάθε πλευράς και τα hints θεωρούνται μεταβλητά, ενώ οι κανόνες και η λογική του παζλ παραμένουν ίδια με το original. Στον χρήστη του προγράμματος Python δίνεται η δυνατότητα να ορίσει όλα τα παραπάνω δεδομένα.

Η μελέτη βιβλιογραφικών πηγών κρίθηκε απαραίτητη τόσο για την θεωρητική κατανόηση του μοντέλου του προβλήματος, όσο και για την υλοποίηση και την επίλυσή του με κώδικα. Αρχικά, η κατανόηση και μελέτη του μοντέλου MILP βασίστηκε σε επιστημονικό άρθρο [2], ενώ η άντληση γενικών πληροφοριών για το παζλ πραγματοποιήθηκε από σχετικές πηγές στο διαδίκτυο [3, 4]. Για την ανάπτυξη του προγράμματος, μελετήθηκαν ήδη υλοποιημένα προγράμματα [5 - 8]. Περαιτέρω πηγές που χρησιμοποιήθηκαν αποκλειστικά για την συγγραφή του κώδικα, για παράδειγμα για την επίλυση κάποιου error, αναφέρονται στα αντίστοιχα σημεία στον κώδικα.

II. ΑΝΑΠΤΥΞΗ ΘΕΜΑΤΟΣ

Στην παρούσα ενότητα αναλύεται η μεθοδολογία που ακολουθήθηκε για την επίλυση του προβλήματος. Ειδικότερα, αναλύονται τα δεδομένα και τα ζητούμενα του προβλήματος και πώς αυτά κωδικοποιούνται και συμβολίζονται ώστε να περιγραφούν από το μαθηματικό μοντέλο. Εξηγείται με λεπτομέρεια η λογική του μοντέλου, το οποίο αποτελείται από τις μεταβλητές απόφασης, την αντικειμενική συνάρτηση και τους περιορισμούς. Επίσης, παρουσιάζεται ο τρόπος με τον οποίο μπορούν να ενσωματωθούν τα hint pieces σε αυτό. Τέλος, παρουσιάζονται τα αποτελέσματα του κώδικα που αναπτύχθηκε για την επίλυση του μαθηματικού μοντέλου από υπολογιστή και σημειώνονται παρατηρήσεις όσον αφορά την επίδοσή του ανάλογα με το μέγεθος του προβλήματος.

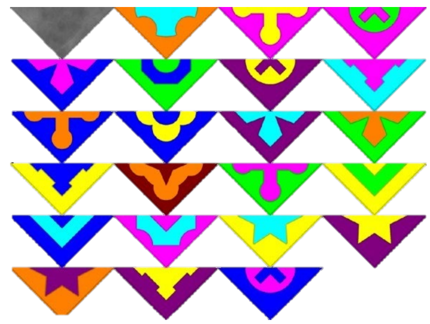
A. Περιγραφή προβλήματος

Το EII αποτελείται από ένα ταμπλό 16×16 θέσεων και από 256 τετράγωνα κομμάτια. Κάθε κομμάτι είναι χωρισμένο σε 4 ισοσκελή τρίγωνα βάσει των δύο διαγωνίων του, αντιστοιχίζοντας έτσι κάθε τρίγωνο σε μία πλευρά του κομματιού. Κάθε τρίγωνο-πλευρά περιέχει έναν συνδυασμό σχημάτων και χρωμάτων. Στο original EII υπάρχουν συνολικά 23 δυνατοί συνδυασμοί. Στην Εικ. 1 απεικονίζεται η συσκευασία του παζλ, στην Εικ. 2 το ταμπλό του, ενώ στην Εικ. 3 απεικονίζονται ενδεικτικά 2 από τα κομμάτια του. Στην Εικ. 4 φαίνονται όλοι οι συνδυασμοί.

Εικ. 1: η συσκευασία του ΕΙΙ [9].

Εικ. 2: το ταμπλό του παζλ με κομμάτια [10].

Εικ. 3: τυχαία κομμάτια του παζλ.



Εικ. 4: όλοι οι δυνατοί συνδυασμοί σχημάτων και χρωμάτων για κάθε πλευρά των κομματιών [11].

Εξαίρεση αποτελούν οι πλευρές που είναι γεμισμένες μόνο με γκρι χρώμα. Αυτές ανήκουν στην περίμετρο του ταμπλό και τα κομμάτια στα οποία ανήκουν, που ονομάζονται εξωτερικά κομμάτια, πρέπει να τοποθετηθούν στις ακριανές θέσεις του. Εάν ένα εξωτερικό κομμάτι έχει 2 γκρι πλευρές, τοποθετείται σε μία από τις 4 γωνίες του ταμπλό, ενώ αν έχει μόνο 1 γκρι πλευρά τότε τοποθετείται σε κάποια άλλη περιμετρική θέση. Εάν ένα κομμάτι δεν έχει καμία γκρι πλευρά, τότε ονομάζεται εσωτερικό κομμάτι και τοποθετείται στο εσωτερικό τμήμα του ταμπλό, δηλαδή μέσα από την περίμετρο. Στην Εικ. 5 απεικονίζονται ενδεικτικά ένα γωνιακό και ένα απλό εξωτερικό κομμάτι.

Εικ. 5: τυχαία εξωτερικά κομμάτια του παζλ.

Το ζητούμενο του παζλ είναι ο παίκτης να ενώσει όλα τα κομμάτια πάνω στο ταμπλό ώστε οι γειτνιάζουσες πλευρές τους να ταιριάζουν, δηλαδή να έχουν ίδιο συνδυασμό σχημάτων και χρωμάτων. Παράλληλα, πρέπει να φροντίσει τα εξωτερικά κομμάτια να είναι τοποθετημένα στην περίμετρο του ταμπλό με τέτοιο προσανατολισμό ώστε οι γκρι πλευρές να βρίσκονται στην εξωτερική μεριά, δημιουργώντας έτσι ένα περίγραμμα του ταμπλό. Στην Εικ. 6 απεικονίζεται ένα σύνολο κομματιών με πλευρές που ταιριάζουν μεταξύ τους, ενώ στην Εικ. 7 φαίνονται πλευρές που δεν ταιριάζουν.

Εικ. 6: παράδειγμα κομματιών του παζλ με πλευρές που ταιριάζουν.

Εικ. 7: παράδειγμα κομματιών του παζλ με κάποιες πλευρές που δεν ταιριάζουν, σημειωμένες με κόκκινο χρώμα.

Σημειώνεται ότι οι υπάρχουσες μαθηματικές μελέτες για το EII δεν συμφωνούν με σιγουριά όσον αφορά την ύπαρξη μίας και μοναδικής λύσης του. Επομένως, ενδέχεται να υπάρχουν πολλαπλές βέλτιστες λύσεις, δηλαδή διαφορετική τοποθέτηση και περιστροφή των κομματιών που να πληρούν τους παραπάνω περιορισμούς.

Στο φυλλάδιο οδηγιών του EII δίνεται ένα hint για την λύση του, δηλαδή αποκαλύπτεται η θέση και ο προσανατολισμός ενός κομματιού από τα 256 πάνω στο ταμπλό. Μαζί με το original παζλ, διατέθηκαν στην αγορά 4 clue παζλ, τα οποία δίνουν ακόμα 4 hints, αυξάνοντας τον συνολικό αριθμό των hints στα 5. Σύμφωνα με το φυλλάδιο οδηγιών του παζλ, δεν είναι απαραίτητη η χρήση των hints για την εύρεση λύσης. Μάλιστα, εάν όντως υπάρχουν πολλαπλές λύσεις τότε τα hints ενδέχεται σε κάποιες από αυτές να μην έχουν την ενδεδειγμένη θέση και προσανατολισμό, καθώς ο συνδυασμός τους πιθανόν να ανήκει σε μία μόνο λύση. Τα clue puzzles έχουν τους ίδιους κανόνες και την ίδια λογική με το original, με τη μόνη διαφορά ότι έχουν μικρότερο μέγεθος (τα 2 είναι 6×6 και τα υπόλοιπα 2 είναι 12×6) και λιγότερους δυνατούς συνδυασμούς σχημάτων και χρωμάτων για την κάθε πλευρά. Στην Εικ. 8 απεικονίζεται μία από τις λύσεις του clue puzzle 1.

Εικ. 8: μία από τις λύσεις του clue puzzle 1 [6].

B. Κωδικοποίηση πληροφοριών και συμβολισμοί

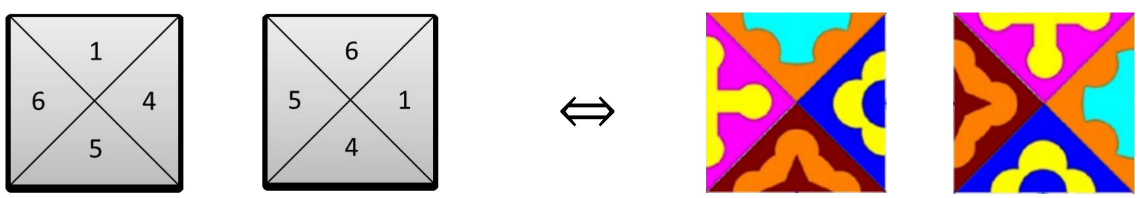
Η βιβλιογραφία που μελετήθηκε για την λύση του EII χρησιμοποιεί συγκεκριμένους συμβολισμούς και κωδικοποιήσεις των δεδομένων του παζλ, για καλύτερη διαχείριση και οργάνωση αυτών. Επιλέχθηκε να ακολουθηθούν οι ίδιοι συμβολισμοί στην εργασία, τόσο στο μαθηματικό μοντέλο όσο και στον κώδικα για το πρόγραμμα.

Αρχικά, τα Eternity-like παζλ θεωρούνται ότι έχουν ταμπλό διαστάσεων n×m, όπου n είναι οι γραμμές και m είναι οι στήλες. Για την αναφορά σε μία γραμμή χρησιμοποιείται το σύμβολο r (row), το οποίο παίρνει ακέραιες τιμές από 1 έως και n, ενώ για την αναφορά σε μία στήλη χρησιμοποιείται το c (column), που παίρνει ακέραιες τιμές από 1 έως και m.

Το παζλ έχει n×m κομμάτια, στα οποία δίνεται ένας μοναδικός ακέραιος αριθμός ως id για να ξεχωρίζουν μεταξύ τους. Για την αναφορά σε ένα κομμάτι χρησιμοποιείται το σύμβολο t (tile), το οποίο παίρνει τιμές από το 1 μέχρι και το (n×m).

Επιπλέον, τα χρώματα και τα σχήματα κάθε πλευράς ενός κομματιού μπορούν να είναι οποιαδήποτε από μία δεδομένη συλλογή σχημάτων και χρωμάτων. Για ευκολότερη διαχείρισή τους, δίνεται σε κάθε συνδυασμό ένας μοναδικός ακέραιος αριθμός ως id. Συνηθίζεται στα Eternity-like παζλ το πλήθος των δυνατών συνδυασμών να συμβολίζεται με L, ενώ για την αναφορά σε έναν συνδυασμό χρησιμοποιείται το l, που παίρνει ακέραιες τιμές από το 0 μέχρι και το (L-1). Η αρίθμηση ξεκινάει από το 0 λόγω της σύμβασης ο συνδυασμός με αριθμό 0 να είναι το γκρι, που αντιστοιχεί στις ακριανές πλευρές των εξωτερικών κομματιών.

Τέλος, κάθε κομμάτι μπορεί να έχει έναν από 4 δυνατούς προσανατολισμούς, αφού είναι τετράγωνο. Για την αναφορά στον προσανατολισμό ενός κομματιού χρησιμοποιείται το σύμβολο a, το οποίο παίρνει ακέραιες τιμές από 0 έως και 3. Η τιμή του a εκφράζει το πόσες φορές έχει περιστραφεί το κομμάτι, με τη φορά του ρολογιού, σε σχέση με την αρχική του κατάσταση. Για παράδειγμα, στην Εικ. 9 στα αριστερά φαίνεται ένα τυχαίο κομμάτι πριν και μετά την περιστροφή του, κατά 90° (μία περιστροφή), ενώ στα δεξιά απεικονίζεται η αλλαγή στο πραγματικό κομμάτι του παζλ.



a = 0 a = 1 a = 0 a = 1

Εικ. 9: στα αριστερά ένα τυχαίο κομμάτι πριν (a = 0) και μετά (a = 1) την περιστροφή του κατά 90°. Κάτω από κάθε εικόνα δίνεται η τιμή του a. Οι τιμές εντός των τριγώνων στο κομμάτι είναι οι τιμή του l της κάθε πλευράς. Στα δεξιά φαίνεται η αλλαγή στο πραγματικό κομμάτι.

C. Μαθηματικό μοντέλο

Για τη μαθηματική περιγραφή του παζλ ως πρόβλημα βελτιστοποίησης, δημιουργήθηκε ένα μοντέλο MILP (Mixed Integer Linear Programming), το οποίο μετατρέπει τα φυσικά δεδομένα και τους περιορισμούς του EII σε μία συνάρτηση και εξισο-ανισώσεις. Το μοντέλο που υλοποιείται αποτελείται από τις μεταβλητές απόφασης, την αντικειμενική συνάρτηση και τους περιορισμούς.

Για την επίλυση του παζλ, εξετάζεται κάθε πλευρά όλων των κομματιών του. Επομένως, είναι αναγκαία η κωδικοποίηση των πληροφοριών κάθε κομματιού και ο συμβολισμός του. Συνοπτικά, όπως εξηγήθηκε στην προηγούμενη ενότητα, θεωρώντας ένα Eternity-like παζλ με ταμπλό n×m διαστάσεων και με πλήθος δυνατών συνδυασμών L για κάθε πλευρά, για την ολοκληρωμένη αναφορά σε μία πλευρά ενός κομματιού απαιτούνται:

- Η θέση του κομματιού, που αποτελείται από την γραμμή r = 1…n και την στήλη c = 1…m
- Το id του κομματιού, που συμβολίζεται με t = 1…(n×m)
- Ο συνδυασμός σχήματος και χρωμάτων της πλευράς, που συμβολίζεται με l = 0…(L-1)

- Οι φορές που έχει περιστραφεί το κομμάτι, που συμβολίζεται με a = 0…3

Για την αναφορά σε ένα κομμάτι, αλλά όχι σε κάποια πλευρά του, απαιτούνται όλες οι παραπάνω πληροφορίες εκτός από τους συνδυασμούς των πλευρών του. Έτσι, ορίζεται η δυαδική μεταβλητή απόφασης $x_{t,r,c,a}$ η οποία παίρνει τιμή 1 αν το κομμάτι t είναι τοποθετημένο στην θέση (r,c) με περιστροφή a.

Για τον έλεγχο αν όλες οι πλευρές ταιριάζουν με τις γειτονικές τους στον οριζόντιο άξονα, ορίζεται η δυαδική μεταβλητή απόφασης $h_{r,c}$ η οποία παίρνει τιμή 1 αν η δεξιά πλευρά της θέσης (r,c) του ταμπλό είναι unmatched, δηλαδή αν οποιοδήποτε κομμάτι σε αυτή την θέση δεν ταιριάζει με το γειτονικό του στην δεξιά του πλευρά.

Αντίστοιχα, για τον έλεγχο αν όλες οι πλευρές ταιριάζουν με τις γειτονικές τους στον κάθετο άξονα, ορίζεται η δυαδική μεταβλητή απόφασης $v_{r,c}$ η οποία παίρνει τιμή 1 αν η κάτω πλευρά της θέσης (r,c) του ταμπλό είναι unmatched, δηλαδή αν οποιοδήποτε κομμάτι σε αυτή την θέση δεν ταιριάζει με το γειτονικό του στην κάτω του πλευρά.

Οι μεταβλητές απόφασης του προβλήματος, οπότε, συνοψίζονται ως εξής:

$$x_{t,r,c,a} = \begin{cases} 1, \text{αν το κομμάτι t είναι τοποθετημένο στην θέση (r, c) με περιστροφή a} \\ 0, \text{αλλιώς} \end{cases}$$

$$h_{r,c} = \begin{cases} 1, \text{αν η δεξιά πλευρά της θέσης (r, c) του ταμπλό είναι unmatched} \\ 0, \text{αλλιώς} \end{cases}$$

$$v_{r,c} = \begin{cases} 1, \text{αν η κάτω πλευρά της θέσης (r, c) του ταμπλό είναι unmatched} \\ 0, \text{αλλιώς} \end{cases}$$

Στόχος του μοντέλου είναι η ελαχιστοποίηση των unmatched πλευρών, δηλαδή η τοποθέτηση και περιστροφή όλων των κομματιών στο ταμπλό έτσι ώστε όλες οι πλευρές να ταιριάζουν μεταξύ τους. Λαμβάνοντας υπόψη ότι το άθροισμα των μεταβλητών $h_{r,c}$ και $v_{r,c}$ για κάθε θέση του ταμπλό είναι το σύνολο όλων των unmatched κομματιών, προκύπτει ότι η αντικειμενική συνάρτηση είναι η:

$$\min \sum_{r=1}^{n} \sum_{c=1}^{m-1} h_{r,c} + \sum_{r=1}^{n-1} \sum_{c=1}^{m} v_{r,c} \tag{1}$$

Ο λόγος που στο άθροισμα των $h_{r,c}$ η αναζήτηση για unmatched πλευρές τερματίζει στην στήλη (m-1) και όχι στην m είναι επειδή εξετάζεται αν η **δεξιά** πλευρά κάθε κομματιού ταιριάζει με την γειτονική της. Επομένως, τα δεξιότερο (εξωτερικό) κομμάτι κάθε γραμμής πρέπει να αγνοηθεί, καθώς η δεξιά πλευρά της βρίσκεται στην περίμετρο του ταμπλό και άρα δεν έχει άλλο κομμάτι με το οποίο γειτονεύει και ταιριάζει.

Αντίστοιχα, στο άθροισμα των $v_{r,c}$ η αναζήτηση σταματά στην γραμμή (n-1), διότι οι κάτω πλευρές των κομματιών στην τελευταία γραμμή ανήκουν στην περίμετρο.

Σημειώνεται ότι εφόσον εξετάζονται οι δεξιές και κάτω πλευρές των κομματιών, παραλείπονται τα εξωτερικά κομμάτια στην πρώτη στήλη και γραμμή, καθώς οι εξωτερικές τους πλευρές δεν εξετάζονται όπως έχουν οριστεί οι $h_{r,c}$ και $v_{r,c}$.

Ωστόσο, η ελαχιστοποίηση των unmatched κομματιών δεν είναι το μόνο ζητούμενο του παζλ, καθώς υπάρχει πληθώρα κανόνων που περιορίζουν το πώς μπορούν να τοποθετηθούν τα κομμάτια. Όλοι αυτοί οι κανόνες μεταφράζονται σε περιορισμούς για το μοντέλο.

Αρχικά, ισχύει ο προφανής κανόνας ότι κάθε κομμάτι του παζλ έχει μία μόνο θέση στο ταμπλό, με έναν συγκεκριμένο προσανατολισμό. Δηλαδή, το άθροισμα των κομματιών με μία συγκεκριμένη θέση και προσανατολισμό πρέπει να είναι ακριβώς ίσο με 1:

$$\sum_{r=1}^{n}\sum_{c=1}^{m}\sum_{a=0}^{3} x_{t,r,c,a} = 1 \qquad \forall\, t = 1, \dots, (\mathrm{n} \times \mathrm{m}) \tag{2}$$

Αντίστοιχα, σε κάθε θέση του ταμπλό πρέπει να τοποθετηθεί ακριβώς ένα κομμάτι:

$$\sum_{t=1}^{\mathrm{n}\times\mathrm{m}}\sum_{a=0}^{3} x_{t,r,c,a} = 1 \qquad \forall\, r = 1, \dots, \mathrm{n}\,, \mathrm{c} = 1, \dots, \mathrm{m} \tag{3}$$

Η διαφορά των περιορισμών (2) και (3) είναι ότι ο (2) περιγράφει την φύση των κομματιών, δηλαδή επιβάλλει κάθε κομμάτι να τοποθετείται σε μία μόνο θέση στο ταμπλό με έναν μόνο προσανατολισμό. Ο περιορισμός (3) περιγράφει την φύση του ταμπλό, δηλαδή επιβάλλει σε κάθε θέση του (γραμμή και στήλη) να υπάρχει ένα μόνο κομμάτι με έναν προσανατολισμό.

Για τον έλεγχο της συμβατότητας των πλευρών των κομματιών, πριν δοθούν οι αντίστοιχοι περιορισμοί του μοντέλου, είναι αναγκαίο να οριστούν οι συντελεστές συμβατότητας (matching coefficients). Ο συντελεστής $CT_{t,a,l}$ είναι ίσος με 1 αν το κομμάτι t έχει συνδυασμό l στην πάνω πλευρά του, όταν περιστραφεί a φορές. Ίδιοι είναι οι ορισμοί των συντελεστών CB, CL, CR για την κάτω, αριστερή και δεξιά πλευρά αντίστοιχα.

Οι συντελεστές αυτοί είναι χρήσιμοι, καθώς όταν ελέγχεται αν δύο πλευρές ταιριάζουν, για παράδειγμα αν η δεξιά πλευρά ενός κομματιού A (στην στήλη c) ταιριάζει με την αριστερή του γειτονικού του B (στην στήλη c+1), τότε εξετάζεται ο συντελεστής CR για την δεξιά πλευρά και ο CL για την αριστερή. Εάν και οι δύο συντελεστές είναι 1, τότε οι δύο πλευρές ταιριάζουν, ενώ αν ένας από τους δύο είναι 0 τότε δεν ταιριάζουν.

Οπότε, το γινόμενο του συντελεστή CR του κομματιού A, στην θέση (r,c), με το $x_{t,r,c,a}$ είναι ίσο με 1 μόνο όταν το κομμάτι t βρίσκεται στην θέση (r,c) με περιστροφή a και έχει ένα συγκεκριμένο χρώμα στην εξεταζόμενη πλευρά του. Δηλαδή, το $x_{t,r,c,a}$ ενεργοποιεί και απενεργοποιεί το σωστό κομμάτι και την περιστροφή του. Το ίδιο ισχύει και για το γινόμενο CL του κομματιού B, στην θέση (r,c+1), με το $x_{t,r,c,a}$ του.

Με την αφαίρεση, επομένως, των δύο γινομένων (A-B), ελέγχεται αν ταιριάζουν η δεξιά πλευρά του A με την αριστερή του B. Αν ταιριάζουν, τότε η διαφορά τους είναι 0, ενώ αν δεν ταιριάζουν θα είναι 1. Επεκτείνοντας αυτή την λειτουργία για κάθε θέση του ταμπλό, λαμβάνεται ο περιορισμός του μοντέλου:

$$\sum_{t=1}^{\mathrm{n}\times\mathrm{m}}\sum_{a=0}^{3} CR_{t,a,l}\cdot x_{t,r,c,a} - \sum_{t=1}^{\mathrm{n}\times\mathrm{m}}\sum_{a=0}^{3} CL_{t,a,l}\cdot x_{t,r,c+1,a} \leq\; h_{r,c} \tag{4}$$
$$\forall\, r = 1, \dots, \mathrm{n}\,, \mathrm{c} = 1, \dots, (\mathrm{m}-1), \mathrm{l} = 1, \dots, (\mathrm{L}-1)$$

Ο λόγος που απαιτείται η διαφορά των γινομένων να είναι μικρότερη ή ίση του $h_{r,c}$ είναι επειδή το $h_{r,c}$ σε περίπτωση που όλες οι πλευρές ταιριάζουν, θα έχει τιμή 0, επομένως η διαφορά των γινομένων πρέπει να είναι 0. Σε περίπτωση όμως που δεν είναι δυνατόν να ταιριάξουν όλα τα κομμάτια μεταξύ τους, για παράδειγμα αν οι πληροφορίες των κομματιών είναι λάθος ή αν οι συνδυασμοί δεν ταιριάζουν ποτέ, τότε δεν υπάρχει ολοκληρωμένη λύση για το παζλ και άρα το μοντέλο πρέπει να υπολογίσει την βέλτιστη, δηλαδή την λύση που θα ταιριάξει όσες περισσότερες πλευρές γίνεται. Λόγω του ότι αναγκαστικά θα υπάρχουν πλευρές που δεν θα ταιριάζουν μεταξύ τους σε κάποιες θέσεις, το $h_{r,c}$ θα έχει τιμή 1 αν υπάρχουν πλευρές στον οριζόντιο άξονα (δεξιές ή αριστερές δηλαδή) που δεν ταιριάζουν με τις γειτονικές τους. Απαιτώντας, λοιπόν, η διαφορά των γινομένων να είναι μικρότερη από 1, απαιτείται σε κάθε θέση να υπάρχει το πολύ μία unmatched πλευρά, δηλαδή στην χειρότερη περίπτωση κάθε κομμάτι να έχει την δεξιά του πλευρά unmatched με την αριστερή του γειτονικού του.

Εφαρμόζοντας τον ίδιο έλεγχο, αυτή τη φορά όμως εξετάζοντας αν η **αριστερή** πλευρά κάθε κομματιού ταιριάζει με την δεξιά της γειτονικής της, προκύπτει ο περιορισμός:

$$-\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CR_{t,a,l}\cdot x_{t,r,c,a} + \sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CL_{t,a,l}\cdot x_{t,r,c+1,a} \le h_{r,c} \tag{5}$$
$$\forall\, r = 1,\dots,\mathrm{n}\,,\mathrm{c} = 1,\dots,(\mathrm{m}-1),\mathrm{l} = 1,\dots,(\mathrm{L}-1)$$

Επεκτείνοντας ακριβώς την ίδια λογική για τον έλεγχο των κάτω πλευρών των κομματιών με τις πάνω των γειτονικών τους, και το αντίστροφο, λαμβάνονται οι περιορισμοί:

$$\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CB_{t,a,l}\cdot x_{t,r,c,a} - \sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CT_{t,a,l}\cdot x_{t,r+1,c,a} \le v_{r,c} \tag{6}$$
$$\forall\, r = 1,\dots,(\mathrm{n}-1),\mathrm{c} = 1,\dots,\mathrm{m}\,,\mathrm{l} = 1,\dots,(\mathrm{L}-1)$$

και

$$-\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CB_{t,a,l}\cdot x_{t,r,c,a} + \sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CT_{t,a,l}\cdot x_{t,r+1,c,a} \le v_{r,c} \tag{7}$$
$$\forall\, r = 1,\dots,(\mathrm{n}-1),\mathrm{c} = 1,\dots,\mathrm{m}\,,\mathrm{l} = 1,\dots,(\mathrm{L}-1)$$

Σε όλους τους παραπάνω περιορισμούς για τον έλεγχο συμβατότητας των πλευρών των κομματιών αγνοείται ο συνδυασμός 0 για το χρώμα των πλευρών. Αυτό συμβαίνει διότι ο συνδυασμός 0 είναι η γκρι πλευρά, η οποία ανήκει στην περίμετρο του ταμπλό. Παρακάτω περιγράφονται οι περιορισμοί που αφορούν αποκλειστικά αυτό τον κανόνα του παζλ και ελαφρύνουν τους υπόλοιπους περιορισμούς.

Για την τοποθέτηση των γκρι πλευρών στην περίμετρο του ταμπλό, χρησιμοποιείται ο συντελεστής CT με l=0 για τις πάνω πλευρές όλων των κομματιών στην πρώτη γραμμή (r=0). Απαιτείται σε κάθε θέση της πρώτης γραμμής του ταμπλό να υπάρχει ένα κομμάτι με την πάνω πλευρά του να είναι γκρι. Έτσι, προκύπτει ο περιορισμός για το μοντέλο:

$$\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CT_{t,a,0}\cdot x_{t,0,c,a} = 1 \qquad \forall\ \mathrm{c} = 1,\dots,\mathrm{m} \tag{8}$$

Επεκτείνοντας την ίδια λογική για την κάτω (CB) πλευρά κάθε κομματιού στην τελευταία γραμμή (r=n-1) του ταμπλό:

$$\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CB_{t,a,0}\cdot x_{t,n-1,c,a} = 1 \qquad \forall\ \mathrm{c} = 1,\dots,\mathrm{m} \tag{9}$$

Αντίστοιχα για την αριστερή (CL) πλευρά κάθε κομματιού στην πρώτη στήλη (c=0) του ταμπλό:

$$\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CL_{t,a,0}\cdot x_{t,r,0,a} = 1 \qquad \forall\ \mathrm{r} = 1,\dots,\mathrm{n} \tag{10}$$

Και τέλος για την δεξιά (CR) πλευρά κάθε κομματιού στην τελευταία στήλη (c=m-1) του ταμπλό:

$$\sum_{t=1}^{\mathrm{n\times m}}\sum_{a=0}^{3} CR_{t,a,0}\cdot x_{t,r,m-1,a} = 1 \qquad \forall\ \mathrm{r} = 1,\dots,\mathrm{n} \tag{11}$$

Οι περιορισμοί για τις δυνατές τιμές των μεταβλητών απόφασης, εφόσον είναι όλες δυαδικές, είναι οι εξής:

$$0 \leq x_{t,r,c,a} \leq 1 \qquad \forall\, t = 1, \dots, (\mathrm{n} \times \mathrm{m}), \mathrm{r} = 1, \dots, \mathrm{n}\,, \mathrm{c} = 1, \dots, \mathrm{m}, \mathrm{a} = 0, \dots, 3 \tag{12}$$

$$0 \leq h_{r,c} \leq 1 \qquad \forall\, \mathrm{r} = 1, \dots, \mathrm{n}\,, \mathrm{c} = 1, \dots, (\mathrm{m} - 1) \tag{13}$$

$$0 \leq v_{r,c} \leq 1 \qquad \forall\, \mathrm{r} = 1, \dots, (\mathrm{n} - 1)\,, \mathrm{c} = 1, \dots, \mathrm{m} \tag{14}$$

Σημειώνεται ότι θα ήταν δυνατό να υπάρχει ένας ακόμα περιορισμός για το μοντέλο που να υποχρεώνει την αντικειμενική συνάρτηση (1) να πάρει τιμή 0, δηλαδή να μην υπάρχουν unmatched πλευρές. Κάτι τέτοιο θα μετέτρεπε το πρόβλημα από πρόβλημα εύρεσης βέλτιστης λύσης με τις λιγότερες unmatched πλευρές σε πρόβλημα εύρεσης εφικτής λύσης με όλες τις πλευρές να ταιριάζουν. Το μοναδικό κέρδος αυτής της εναλλακτικής προσέγγισης είναι ότι στον υπολογιστή ενδέχεται να βρεθεί γρηγορότερα αποτέλεσμα σε σχέση με την επιλεγμένη προσέγγιση, μιας και ο αποκλεισμός των περιπτώσεων συμβαίνει πιο σύντομα, αμέσως μόλις παραβιαστεί ο περιορισμός. Επιλέχθηκε να μην ακολουθηθεί αυτή η λογική καθώς στην γενική περίπτωση των Eternity-like παζλ, στα πλαίσια των οποίων ο χρήστης παρέχει στο μοντέλο όλα τα δεδομένα, δεν είναι εγγυημένο ότι υπάρχει λύση που να ταιριάζει όλες τις πλευρές, οπότε η εναλλακτική προσέγγιση σε αυτή την περίπτωση δεν θα επέστρεφε λύση.

D. Hint pieces

Θεωρώντας ότι σε ένα Eternity-like παζλ υπάρχει ένα σύνολο στοιχείων (hint pieces) για την λύση του, είναι δυνατό αυτά να ενσωματωθούν στο μαθηματικό μοντέλο που το περιγράφει με τη μορφή περιορισμών. Λαμβάνοντας υπόψη ότι ένα hint ορίζεται ως η θέση και ο προσανατολισμός ενός δεδομένου κομματιού, οι περιορισμοί θα πρέπει να επιβάλλουν την τοποθέτηση και την περιστροφή του κάθε hint όπως δίνονται από τα δεδομένα.

Στην γενική περίπτωση, θεωρώντας ένα πλήθος H hints, μπορεί να χρησιμοποιηθεί ο δείκτης h για την αναφορά σε καθένα από αυτά. Επομένως, για ένα τυχαίο hint δίνεται:

- Το id του, δηλαδή ποιο κομμάτι είναι: t_h
- Η θέση του στο ταμπλό: (r_h, c_h)
- Η περιστροφή του: a_h

με h = 1,…,H .

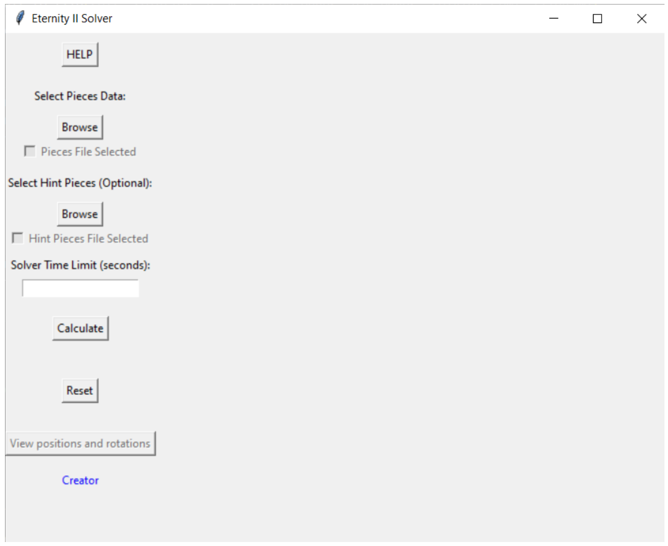
Ο κανόνας για την επιβολή της τοποθέτησης των hints στις δεδομένες θέσεις με τους προσανατολισμούς τους μπορεί να γραφεί ως:

$$x_{t_h, r_h, c_h, a_h} = 1 \qquad \forall\, h = 1, \dots, \mathrm{H} \tag{15}$$

E. Πρόγραμμα Python

Για την πρακτική εφαρμογή του MILP μοντέλου, αναπτύχθηκε πρόγραμμα σε Python που υλοποιεί το μοντέλο και λύνει το πρόβλημα. Αρχικά, ο χρήστης παρέχει στο πρόγραμμα ένα αρχείο με τα δεδομένα του παζλ που θέλει να λύσει και δίνει προαιρετικά τα hint pieces και το μέγιστο χρόνο για τον οποίο θα προσπαθήσει το πρόγραμμα να λύσει το πρόβλημα. Ύστερα, υπολογίζεται η βέλτιστη λύση και εμφανίζεται στον χρήστη. Παρακάτω εξηγούνται με λεπτομέρεια όλα τα παραπάνω βήματα λειτουργίας, συνοδευόμενα από παραδείγματα χρήσης της εφαρμογής και screenshots.

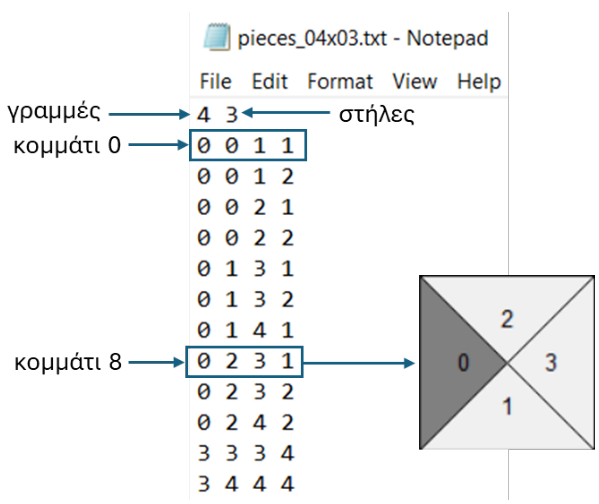
Για τον σκοπό της εργασίας, αναπτύχθηκε μία γραφική διεπαφή (GUI), με την οποία ο χρήστης αλληλεπιδρά. Η οθόνη που εμφανίζεται με το άνοιγμα της εφαρμογής φαίνεται στην Εικ. 10.



Εικ. 10: το GUI του προγράμματος κατά το άνοιγμά του.

Αρχικά, ο χρήστης καλείται να παρέχει στο πρόγραμμα τα δεδομένα του παζλ (pieces data) που επιθυμεί να λύσει. Συγκεκριμένα, ζητείται η επιλογή για άνοιγμα ενός txt αρχείου, το οποίο περιέχει τις διαστάσεις του ταμπλό, δηλαδή τις γραμμές και τις στήλες, καθώς και τα δεδομένα κάθε κομματιού, δηλαδή τον συνδυασμό κάθε πλευράς κάθε κομματιού.

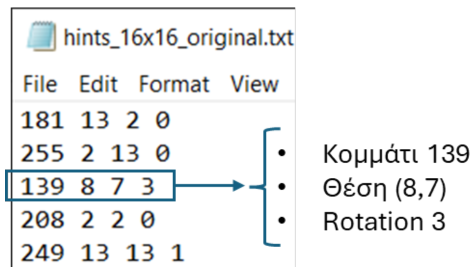
Για την επεξεργασία όλων των πληροφοριών, απαιτείται αυτές να είναι αποθηκευμένες στο txt αρχείο με συγκεκριμένο format. Ειδικότερα, στην πρώτη γραμμή πρέπει να υπάρχουν μόνο δύο αριθμοί χωρισμένοι με κενό, που είναι το πλήθος των γραμμών και των στηλών του ταμπλό, ενώ όλες οι υπόλοιπες γραμμές αφορούν τα δεδομένα των κομματιών. Για κάθε κομμάτι αφιερώνεται μία γραμμή του αρχείου, με την κάθε γραμμή να αποτελείται από 4 αριθμούς χωρισμένους με κενό. Ο πρώτος αριθμός είναι ο συνδυασμός της αριστερής πλευράς του κομματιού, ο δεύτερος είναι ο συνδυασμός της πάνω πλευράς, ο τρίτος είναι αυτός της δεξιάς πλευράς και ο τέταρτος είναι αυτός της κάτω πλευράς. Στην Εικ. 11 φαίνονται τα δεδομένα για ένα 4×3 παζλ με σχετικές ενδείξεις που συνοψίζουν τον παραπάνω τρόπο κωδικοποίησης.



Εικ. 11: τα δεδομένα ενός 4×3 παζλ με ενδείξεις που εξηγούν το format. Ο συνδυασμός 0 είναι για τις πλευρές που ανήκουν στην περίμετρο του ταμπλό που παίρνουν γκρι χρώμα.

Ο χρήστης, οπότε, καλείται να επιλέξει ένα αρχείο txt με τα απαραίτητα δεδομένα, κάνοντας κλικ στο κουμπί 'Browse' κάτω από την ένδειξη 'Select Pieces Data' και επιλέγοντας το επιθυμητό αρχείο μέσω του pop-up window που εμφανίζεται. Αφότου φορτωθούν τα δεδομένα, το αντίστοιχο checkbox κάτω από το κουμπί θα τσεκαριστεί αυτόματα, ώστε να ξέρει ο χρήστης αν το παζλ φορτώθηκε με επιτυχία. Μαζί με τον κώδικα του προγράμματος, παρέχονται στον χρήστη 3 σετ δεδομένων για παζλ. Τα 2 σετ (φάκελοι pieces_set_1 και pieces_set_2) από τα 3 περιέχουν ένα σύνολο έτοιμων αρχείων με δεδομένα για παζλ διαστάσεων από 3×3 μέχρι και 20×20, συμπεριλαμβανομένων και των κομματιών για το original EII (16×16) αλλά και για τα clue puzzles (6×6 και 12×6) [8]. Το 3ο σετ (special_set) περιλαμβάνει παζλ τα οποία έχουν ιδιαιτερότητες και εξετάζουν ακραίες περιπτώσεις, η κάθε μία εκ των οποίων αναφέρεται στο όνομα κάθε αρχείου. Για παράδειγμα, σε ένα παζλ περιέχονται περισσότερες εξωτερικές πλευρές από αυτές που χωράνε στο ταμπλό (4x4_more_edges), ενώ σε άλλο με τα κομμάτια του δεν είναι δυνατό να βρεθεί ολοκληρωμένη λύση με όλες του τις πλευρές να ταιριάζουν (4x4_no_solution).

Μόλις ο χρήστης φορτώσει τα δεδομένα του παζλ, έχει τη δυνατότητα να φορτώσει τα δεδομένα των hint pieces, εάν επιθυμεί να τα χρησιμοποιήσει. Τα hint pieces φορτώνονται πάλι μέσω αρχείου txt, αλλά αυτή τη φορά με διαφορετικό format. Για παράδειγμα, εάν ο χρήστης θέλει να φορτώσει A hint pieces, θα πρέπει το txt να έχει A γραμμές (μία γραμμή για κάθε κομμάτι), η καθεμία από τις οποίες να περιέχει 4 αριθμούς χωρισμένους με κενό. Ο πρώτος αριθμός είναι το id του κομματιού, ο δεύτερος και ο τρίτος είναι η θέση του στο ταμπλό (γραμμή και στήλη), ενώ ο τέταρτος είναι η περιστροφή του. Στην Εικ. 12 συνοψίζεται αυτό το format με σχετικές ενδείξεις.



Εικ. 12: τα δεδομένα των hint pieces για το original EII παζλ, με ενδείξεις που εξηγούν το format.

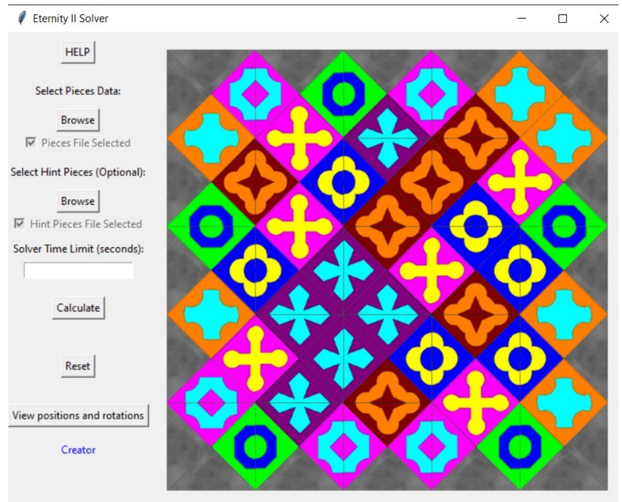
Εάν ο χρήστης επιλέξει αρχείο για hint pieces, τότε το μοντέλο λαμβάνει υπόψη τα δεδομένα και ενεργοποιεί τον περιορισμό (15) για την συμπερίληψή τους στην λύση. Εάν ο χρήστης δεν δώσει hint pieces, τότε ο περιορισμός παραμένει ανενεργός και το πρόγραμμα υπολογίζει την βέλτιστη λύση χωρίς hint pieces. Όπως και για το φόρτωμα των δεδομένων του παζλ, έτσι για το φόρτωμα των hint pieces ο χρήστης μπορεί να κάνει κλικ στο κουμπί 'Browse' κάτω από το label 'Select Hint Pieces' και να επιλέξει το αρχείο μέσω του pop-up window. Αν φορτωθούν τα hint pieces με επιτυχία, τότε το checkbox κάτω από το κουμπί θα τσεκαριστεί αυτόματα.

Όπως αναλύεται στην επόμενη ενότητα, όταν το πρόγραμμα προσπαθεί να λύσει μεγάλων διαστάσεων παζλ, τότε μπορεί να πάρει ώρες μέχρι να παραχθεί λύση. Για αυτό τον λόγο, προαιρετικά ο χρήστης μπορεί να θέσει ένα χρονικό όριο, ώστε το πρόγραμμα να σταματήσει όταν αυτό παρέλθει και να εμφανίσει την καλύτερη λύση που έχει υπολογιστεί μέχρι τότε, σημειώνοντας με κόκκινο χρώμα τις πλευρές που δεν ταιριάζουν. Για να θέσει το χρονικό όριο, ο χρήστης μπορεί να βάλει έναν αριθμό στο entry field με label 'Solver Time Limit', που θα είναι ο χρόνος σε δευτερόλεπτα μέχρι τον οποίο θα επιτραπεί το πρόγραμμα να υπολογίσει την λύση. Εάν το entry field παραμείνει κενό, τότε δεν ορίζεται κάποιο χρονικό όριο και το πρόγραμμα θα τερματίσει μόνο όταν βρεθεί η βέλτιστη λύση.

Αφότου ο χρήστης ορίσει τα δεδομένα του προβλήματος και αν το επιθυμεί τα hint pieces και το χρονικό όριο, με το πάτημα του κουμπιού 'Calculate' το πρόγραμμα δημιουργεί το μαθηματικό μοντέλο και το επιλύει με χρήση του CBC solver. Ο CBC solver επιλέχθηκε λόγω του ότι έρχεται προεγκατεστημένος με την βιβλιοθήκη pulp της Python για τη μοντελοποίηση και επίλυση προβλημάτων γραμμικού προγραμματισμού, και επειδή είναι απλός στην κατανόηση των εντολών του. Μόλις το πρόγραμμα βρει την βέλτιστη λύση, αυτή εμφανίζεται στο solution board στο δεξί μέρος του παραθύρου, δηλαδή δεξιά από τα πεδία και κουμπιά για το φόρτωμα των αρχείων κλπ.

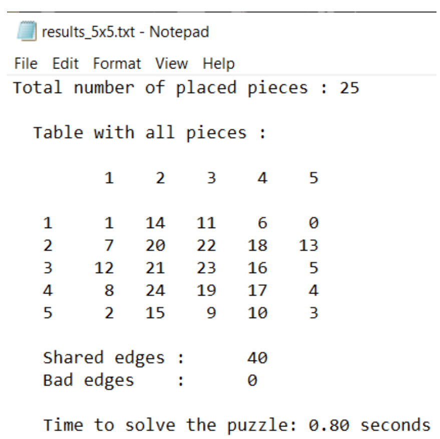
Στην Εικ. 13α φαίνεται ένα παράδειγμα χρήσης του προγράμματος, στο οποίο έχει επιλεγεί η επίλυση ενός 5×5 παζλ με χρονικό όριο 10 δευτερολέπτων, χωρίς hint pieces. Στην Εικ. 13β φαίνεται η λύση του ίδιου παζλ, αλλά αυτή τη φορά χωρίς χρονικό όριο και με τη χρήση hint pieces.

Εικ. 13α: λύση ενός 5×5 παζλ με χρονικό όριο 10 δευτερολέπτων, χωρίς hint pieces.



Εικ. 13β: λύση του ίδιου 5×5 παζλ, αλλά αυτή τη φορά χωρίς χρονικό όριο και με τη χρήση hints.

Τα αποτελέσματα για κάθε παζλ που λύνεται αποθηκεύονται αυτόματα στον φάκελο 'results', σε αρχεία txt. Σε αυτά, εγγράφεται ένα ταμπλό που παρομοιάζει αυτό που σχεδιάζεται στο GUI, με τη μόνη διαφορά ότι στις θέσεις του ταμπλό τοποθετούνται τα ids των κομματιών [6]. Επίσης, κάτω από την λύση αποθηκεύονται ο αριθμός των πλευρών που ταιριάζουν και αυτών που δεν ταιριάζουν, όπως επίσης και ο χρόνος που χρειάστηκε για να λυθεί. Σε περίπτωση που ο χρήστης θέλει να δει άμεσα το id, την θέση και την περιστροφή κάθε κομματιού της λύσης, μπορεί να πατήσει το κουμπί 'View positions and rotations', το οποίο εμφανίζει σε pop-up window αυτές τις πληροφορίες. Για το προηγούμενο παράδειγμα του 5×5 παζλ χωρίς χρονικό όριο και με τη χρήση hints, στην Εικ. 14 δίνεται το αρχείο της λύσης.



```
results_5x5.txt - Notepad
File  Edit  Format  View  Help
Total number of placed pieces : 25

  Table with all pieces :

        1    2    3    4    5

 1      1   14   11    6    0
 2      7   20   22   18   13
 3     12   21   23   16    5
 4      8   24   19   17    4
 5      2   15    9   10    3

   Shared edges :      40
   Bad edges    :      0

   Time to solve the puzzle: 0.80 seconds
```

Εικ. 14: οι πληροφορίες που αποθηκεύονται σε txt αρχείο για την λύση του 5×5 παζλ χωρίς χρονικό όριο και με τη χρήση hints.

Εάν ο χρήστης επιθυμεί να λύσει διαφορετικό παζλ, μπορεί να πατήσει το κουμπί 'Reset', το οποίο σβήνει τα αποθηκευμένα δεδομένα για το παζλ, τα δεδομένα για τα hint pieces και το χρονικό όριο και καθαρίζει το solution board, επιστρέφοντας την οθόνη σε αυτή που φαίνεται στην Εικ. 10.

Τα παραπάνω formats για τα δεδομένα του παζλ, των hint pieces και της λύσης δεν επιλέχθηκαν αυθαίρετα, αλλά βασίστηκαν στα formats που χρησιμοποιούνται από την πλειοψηφία των προγραμμάτων που υπάρχουν στο διαδίκτυο, που λύνουν το ίδιο πρόβλημα.

Τα βήματα για την χρήση του προγράμματος και βοήθεια για τα κουμπιά, τους περιορισμούς και τις λειτουργίες μπορεί να τα διαβάσει άμεσα κανείς πατώντας το κουμπί 'HELP' πάνω-πάνω στο αριστερό μισό του παραθύρου. Με το πάτημα του κουμπιού, εμφανίζεται σχετικό pop-up window.

Ο κώδικας του προγράμματος, καθώς και όλα τα αρχεία που το συνοδεύουν, είναι διαθέσιμα στο link [1]. Ο χρήστης θα πρέπει αρχικά να εγκαταστήσει τις βιβλιοθήκες που αναφέρονται στο αρχείο requirements.txt . Το πρόγραμμα αναπτύχθηκε στην έκδοση της Python 3.10.10, οπότε δεν διασφαλίζεται η σωστή λειτουργία του σε άλλες εκδόσεις.

F. Επιδόσεις προγράμματος

Για τη μέτρηση των επιδόσεων του προγράμματος, χρονομετρήθηκε μόνο η εντολή prob.solve, η οποία δέχεται ως είσοδο το MILP μοντέλο που υλοποιείται για το κάθε παζλ και το λύνει. Το φόρτωμα των αρχείων με τα δεδομένα κάθε παζλ καθώς και η εμφάνιση των αποτελεσμάτων και η αποθήκευσή τους σε αρχεία παρατηρήθηκε ότι παίρνει ελάχιστο χρόνο, της τάξης του 0.5 sec, οπότε αγνοήθηκαν για τη λήψη συμπερασμάτων όσον αφορά το πόσο αποδοτικό είναι το πρόγραμμα.

Για την καταγραφή των μετρήσεων και των τιμών που περιγράφουν την κάθε περίπτωση, υλοποιήθηκε λειτουργία στο πρόγραμμα που εισάγει μία νέα εγγραφή στο αρχείο log. Κάθε εγγραφή αποτελείται από:

1. Το μέγεθος του παζλ (size)
2. Τον αριθμό των hint pieces (hints)
3. Το χρονικό όριο, αν δόθηκε από τον χρήστη (timelimit)
4. Το ποσοστό % ολοκλήρωσης της λύσης (%complete), όπως αυτό υπολογίζεται από τον τύπο:
$$completion = \frac{matched\ edges}{total\ edges} \cdot 100\%$$
5. Τον χρόνο που πήρε να βρεθεί η λύση (time)

Επιλέχθηκε να μην καταγραφεί το μέγιστο πλήθος συνδυασμών για την κάθε εγγραφή, όπως και άλλα μεγέθη, μιας και μία ολοκληρωμένη ανάλυση της κάθε περίπτωσης θα απαιτούσε καταγραφές για ένα μεγάλο σύνολο μετρήσεων, κάτι που είναι εκτός του σκοπού της εργασίας.

Ενδεικτικά, στην Εικ. 15 φαίνεται μέρος του αρχείου log με μετρήσεις.

```
log.txt - Notepad
File  Edit  Format  View  Help
EII SOLVER SOLUTION LOGS

size|hints|timelimit|%complete|time(seconds)
---------------------------------------------
3x3     0       -       100%    0.08
4x3     0       -       100%    0.10
4x4     0       -       100%    0.58
5x3     0       -       100%    0.16
5x4     0       -       100%    0.52
5x5     0       -       100%    52.68
5x5     1       -       100%    29.14
5x5     2       -       100%    14.03
5x5     3       -       100%    0.46
5x5     4       -       100%    0.32
5x5     5       -       100%    0.26
6x3     0       -       100%    0.23
```

Εικ. 15: μέρος του αρχείου log με μετρήσεις.

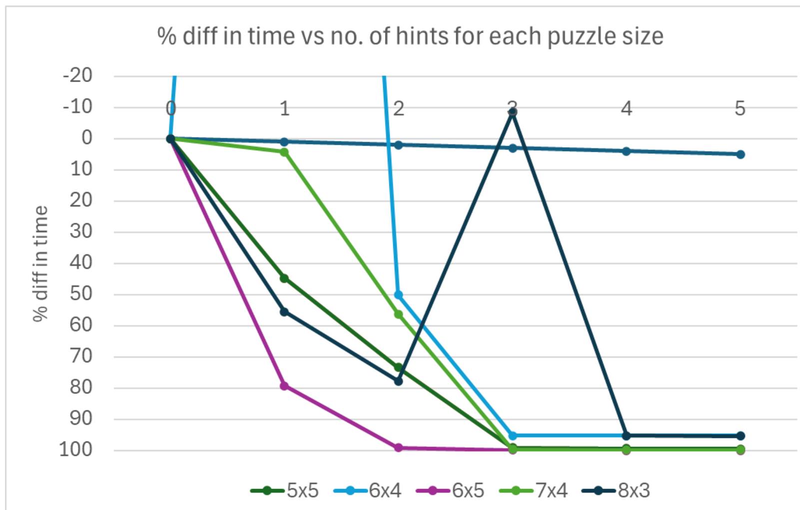
Για παζλ μέχρι και 30 κομμάτια (6×5) ο χρόνος εκτέλεσης δεν ξεπερνά τα 800 seconds. Παρατηρήθηκε ότι έστω και η παραμικρή αύξηση του πλήθους των κομματιών, πάνω από τα 30, επιφέρει κατακόρυφη αύξηση στον χρόνο εκτέλεσης, ξεπερνώντας τις 8 ώρες. Λόγω περιορισμένου διαθέσιμου χρόνου για δοκιμές, τέθηκε ένα άνω όριο στα 30 κομμάτια, δηλαδή το μεγαλύτερο παζλ που δοκιμάστηκε είναι το 6×5.

Για να διασφαλιστεί ότι το περιβάλλον δοκιμής του κάθε παζλ είναι ίδιο, καταγράφηκε για κάθε μέγεθος παζλ ο χρόνος εκτέλεσης συναρτήσει του πλήθους των hints που χρησιμοποιούνται για την λύση του. Ειδικότερα, για κάθε παζλ με χρόνο εκτέλεσης χωρίς hints μεγαλύτερο του 1 second (5×5, 6×4, 6×5, 7×4, 8×3), μετρήθηκε ο χρόνος με εφαρμογή 1 έως και 5 hints.

Για την ανάλυση των μετρήσεων, υπολογίστηκε για το κάθε μέγεθος παζλ η ποσοστιαία διαφορά μεταξύ του χρόνου για το κάθε πλήθος hints από 1 έως 5 hints και του χρόνου για 0 hints. Στην Εικ. 16α φαίνεται η εφαρμογή αυτής της λογικής για τις μετρήσεις του 5×5, ενώ στην Εικ. 16β αναπαρίστανται γραφικά τα παραγόμενα αποτελέσματα για όλα τα μεγέθη παζλ.

size	hints	time	% diff in time
5x5	0	52,68	0,0
5x5	1	29,14	44,7
5x5	2	14,03	73,4
5x5	3	0,46	99,1
5x5	4	0,32	99,4
5x5	5	0,26	99,5

Εικ. 16α: παράδειγμα εφαρμογής της ανάλυσης στο 5×5.



Εικ. 16β: γραφική παράσταση των δεδομένων, μέσω του MS Excel.

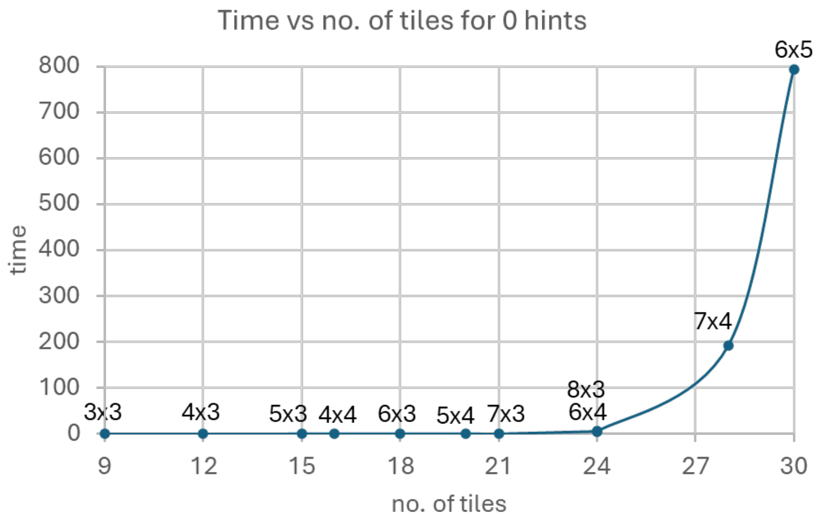
Από την Εικ. 16β είναι φανερό ότι η μεγαλύτερη μεταβολή (δηλαδή η κλίση της) στην ποσοστιαία διαφορά στον χρόνο εκτέλεσης εμφανίζεται με την χρήση 2 hints, στην πλειοψηφία των μεγεθών των παζλ. Ενώ θα περίμενε κανείς, όμως, να αυξάνεται η ποσοστιαία διαφορά με την αύξηση του πλήθους των hints, σε συγκεκριμένες περιπτώσεις παρατηρείται μείωσή της, δηλαδή αύξηση του χρόνου εκτέλεσης με την αύξηση των hints.

Σημειώνεται ότι για την λήψη των παραπάνω συμπερασμάτων δεν λήφθηκε υπόψη το ενδεχόμενο το κάθε hint να έχει διαφορετική επίδραση στην λύση. Δηλαδή, ένα hint που αποκαλύπτει την θέση και περιστροφή ενός γωνιακού κομματιού έχει σίγουρα μικρότερη επίδραση από ένα hint που αποκαλύπτει την θέση ενός εσωτερικού κομματιού, καθώς τα γωνιακά κομμάτια είναι μόνο 4 στο πλήθος και οι δυνατές θέσεις τους στο ταμπλό είναι έτσι κι αλλιώς ελάχιστες (4 γωνίες, άρα 4 δυνατές θέσεις για το καθένα). Λαμβάνοντας υπόψη ότι η επιλογή των hints για το κάθε παζλ έγινε τυχαία, οπότε, ενδέχεται οι μη αναμενόμενες συμπεριφορές των γραφικών παραστάσεων να ευθύνονται στην διαφορετική επίδραση που έχει η προσθήκη του κάθε hint.

Τέλος, μετρήθηκε η επίδραση της αύξησης του μεγέθους του παζλ στον χρόνο εκτέλεσης. Συγκεκριμένα, από το log επιλέχθηκαν οι μετρήσεις για όλα τα μεγέθη παζλ με 0 hints και δημιουργήθηκε ο πίνακας δεδομένων της Εικ. 17α και ακολούθως παράχθηκε το γράφημα της Εικ. 17β.

size	no of tiles	time
3x3	9	0,08
4x3	12	0,1
5x3	15	0,16
4x4	16	0,58
6x3	18	0,23
5x4	20	0,52
7x3	21	0,27
6x4	24	5,53
8x3	24	5,03
7x4	28	191,97
6x5	30	793,72

Εικ. 17α: χρόνοι εκτέλεσης συναρτήσει του μεγέθους του παζλ με 0 hints.



Εικ. 17β: γραφική παράσταση των δεδομένων, μέσω του MS Excel.

Από το γράφημα της Εικ. 17β παρατηρείται κατακόρυφη αύξηση του χρόνου εκτέλεσης, ακόμα και με την παραμικρή αύξηση των διαστάσεων του παζλ. Το φαινόμενο αυτό παρουσιάζεται έντονα με την σύγκριση των χρόνων εκτέλεσης για παζλ 6×5 και 7×4, στην οποία περίπτωση παρόλο που το 6×5 έχει μόνο 2 παραπάνω κομμάτια από το 7×4, έχει χρόνο εκτέλεσης αυξημένο κατά 600 seconds. Ωστόσο, με την σύγκριση των 6×4 και 8×3 παρατηρείται ότι κρατώντας σταθερό τον αριθμό κομματιών, ο χρόνος εκτέλεσης δεν αλλάζει ιδιαίτερα.

Επίσης, μία ακόμα σημαντική παρατήρηση που φαίνεται από την Εικ. 17α είναι ότι τα παζλ με ίσο πλήθος αριθμών και γραμμών, έχουν μεγαλύτερο χρόνο εκτέλεσης σε σχέση με παζλ που μπορεί να έχουν ακόμα και μεγαλύτερο πλήθος κομματιών αλλά διαφορετική κατανομή θέσεων στους άξονες. Αυτό φαίνεται ιδιαίτερα με την σύγκριση των 4×4 και 6×3, στην οποία περίπτωση παρόλο που το 6×3 έχει δύο κομμάτια παραπάνω, παίρνει περισσότερη ώρα να λυθεί.

Το πρόγραμμα αναπτύχθηκε και δοκιμάστηκε σε Laptop με λειτουργικό Windows 10, επεξεργαστή AMD Ryzen 5 @ 2.1GHz και 12GB RAM, χωρίς αξιοποίηση λειτουργίας multithreading ή της GPU. Δεν δοκιμάστηκε σε οποιοδήποτε άλλο σύστημα, οπότε οι επιδόσεις του προγράμματος ενδέχεται να διαφέρουν από αυτές που μετρήθηκαν και παρουσιάστηκαν παραπάνω.

III. ΣΥΝΟΨΗ

Η εργασία αυτή συνδύασε την χρήση ενός συνόλου γνώσεων και μεθόδων που αφορούν τη μαθηματική μοντελοποίηση προβλημάτων βελτιστοποίησης, καθώς και την διατύπωσή τους με τρόπο τέτοιο που να μπορούν να αντιμετωπιστούν από έναν υπολογιστή.

Παρ' ότι περιορισμένη σε εύρος, η εργασία έδωσε μία πολύ χρήσιμη πρώτη επαφή με την πρακτική εφαρμογή της θεωρίας σε ένα πραγματικό πρόβλημα με μεγάλο πλήθος μεταβλητών και περιορισμών. Η μετατροπή των δεδομένων του κάθε Eternity-like παζλ σε ένα πλήρες μοντέλο MILP, το οποίο αποτελείται από την αντικειμενική συνάρτηση, τις μεταβλητές απόφασης και τους περιορισμούς του προβλήματος, ήταν με διαφορά το πιο ενδιαφέρον αλλά και απαιτητικό μέρος της εργασίας. Επιπλέον, ήταν ιδιαίτερα χρήσιμο το γεγονός ότι απαιτήθηκε κάποια έρευνα σε ένα σύνολο πόρων (κυρίως στο διαδίκτυο) για να βρεθεί η πληροφορία σχετικά με την κωδικοποίηση των δεδομένων του παζλ, η οποία είναι απαραίτητη σε κάθε λειτουργία του προγράμματος από την αποθήκευση την κομματιών ενός παζλ μέχρι και την γραφική απεικόνιση και επεξεργασία της λύσης του, καθώς και για να ξεπεραστούν τα προβλήματα που εμφανίστηκαν κατά τη διάρκεια ανάπτυξης του κώδικα.

Τέλος, το πρόγραμμα και οι τεχνικές που χρησιμοποιεί για την επίλυση του προβλήματος σχεδιάστηκαν με απλές προδιαγραφές και λογική, καθώς δόθηκε περισσότερη έμφαση στην κατανόηση του μοντέλου MILP και την σωστή υλοποίησή του με την Python. Επιλέχθηκε να αγνοηθεί το θέμα της επίδοσης, μιας και ήταν προτεραιότητα να σχεδιαστεί μία ολοκληρωμένη γραφική διεπαφή που να επιτρέπει στον χρήστη τον ορισμό κάθε μεγέθους του παζλ και την εξατομίκευση των παραγόμενων αποτελεσμάτων με βάση τις επιθυμίες του. Παρόλο που δεν καταφέρνει να λύσει μεγάλου μεγέθους προβλήματα σε λογικό χρονικό διάστημα, το πρόγραμμα εγγυάται τόσο την σωστή εύρεση βέλτιστης λύσης για μικρά μεγέθη, όσο και τον εντοπισμό και την αντιμετώπιση ακραίων περιπτώσεων, όπως λάθη στα δεδομένα του παζλ.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]. https://github.com/PanosLelakis/EII-MILP-solver

[2]. Fabio Salassa, Wim Vancroonenburg, Tony Wauters, Federico Della Croce, Greet Vanden Berghe, 'MILP and Max-Clique based heuristics for the Eternity II puzzle', 2017, πρόσβαση μέσω του link: https://arxiv.org/abs/1709.00252

[3]. https://en.wikipedia.org/wiki/Eternity_II_puzzle

[4]. https://eu.woodtrick.com/blogs/news/eternity-ii-puzzle-the-hardest-puzzle-in-the-world?SkipCozyRedirect=yes

[5]. https://gitlab.aliens-lyon.fr/rwatriga/eternityii

[6]. http://www.cristal.org/Eternity-II/shorter.html

[7]. https://github.com/TheSil/edge_puzzle/tree/main/data

[8]. https://github.com/david3x3x3/eternity2/tree/master/python

[9]. https://picclick.co.uk/Eternity-2-II-Board-Game-Puzzle-Board-Strategy-295193368888.html

[10]. https://mathsfeedblog.wordpress.com/2022/09/29/who-wants-to-be-a-puzzle-millionaire/

[11]. https://gitlab.aliens-lyon.fr/rwatriga/eternityii/-/blob/master/pieces.png?ref_type=heads