ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Τεχνητή Νοημοσύνη
και
Έμπειρα Συστήματα

6ο εξάμηνο
2022-2023

Εργασία

08/05/2023

Δημήτρης Στυλιανού

Π20004

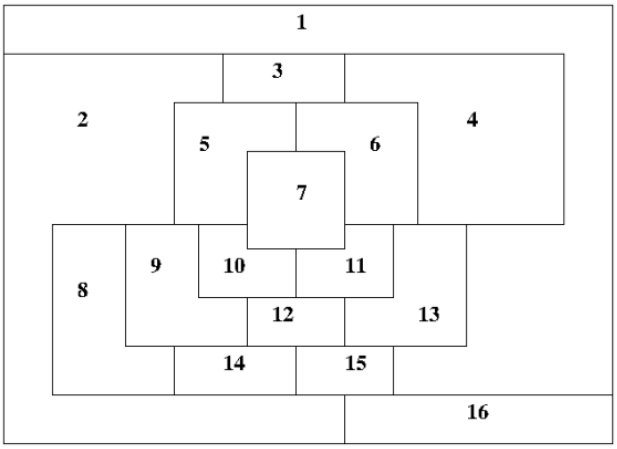
Περιεχόμενο	**Σελίδα**

Εκφώνηση

Θέμα προαιρετικής εργασίας για το μάθημα «Τεχνητή Νοημοσύνη και Έμπειρα Συστήματα». Bonus 2 βαθμοί.

<u>Η εργασία είναι ατομική. Παραδοτέο είναι αρχείο pdf με τεκμηριωμένο κώδικα και παραδείγματα εκτέλεσης, περίπου 5 σελίδων. Υποβολή αποκλειστικά μέσω gunet2. ΜΗ ΣΤΕΙΛΕΤΕ EMAIL. ΜΗ ΣΤΕΙΛΕΤΕ ΠΗΓΑΙΟ ΚΩΔΙΚΑ.</u>

Αναπτύξτε πρόγραμμα χρωματισμού του παρακάτω γράφου με χρήση γενετικών αλγορίθμων και γλώσσα προγραμματισμού της επιλογής σας. Τα διαθέσιμα χρώματα είναι 4: μπλε, κόκκινο, πράσινο, κίτρινο.



Χρησιμοποιείστε τυχαίο αρχικό πληθυσμό με πλήθος της δικής σας επιλογής. Χρησιμοποιείστε συνάρτηση καταλληλότητας και διαδικασία επιλογής γονέων σας της δικής σας επιλογής, επίσης. Χρησιμοποιείστε αναπαραγωγή με διασταύρωση ενός σημείου. Επιλέξτε αν θέλετε να κάνετε και μερική ανανέωση πληθυσμού σε κάποιο ποσοστό π.χ. 30% και μετάλλαξη ενός ψηφίου π.χ. στο 10% του πληθυσμού.

Παραδοτέα της εργασίας είναι ένα pdf (όχι zip, όχι πηγαίος κώδικας) που να περιλαμβάνει τον κώδικα και να τον εξηγεί, να εξηγεί τον τρόπο δράσης του υπολογιστή σύμφωνα με τον αλγόριθμο επίλυσης και να περιλαμβάνει παραδείγματα εκτέλεσης του προγράμματος που αναπτύξατε.

Γενική περιγραφή

Τον κώδικα της εργασίας θα τον βρείτε εδώ.

Αρχικά, στο αρχείο ‘graph.json’ βρίσκεται το λεξικό το οποίο χρησιμοποιείται ως είσοδος για το κυρίως πρόγραμμα ‘genetic_algorithm.py’. Το λεξικό περιέχει τους κόμβους του δοθέντα γράφου, καθώς και τους γειτονικούς κόμβους κάθε κόμβου και είναι της μορφής:

{

node_1: [neighbor_1, ..., neighbor_ j],

.

.

.

node_i: [neighbor_1, ..., neighbor_ j],

.

.

.

node_n: [neighbor_1, ..., neighbor_ j]

}

όπου n είναι το πλήθος κόμβων, node_i (1<=i<=n) αντιστοιχεί σε έναν από τους n κόμβους του γράφου και neighbor_ j (1<=j<=n) οι γειτονικοί κόμβοι του node_i.

Ακολουθεί εν συντομία τα βήματα του γενετικού αλγόριθμου που υλοποιήθηκε:

1. Το αρχικό πλήθος πληθυσμού επιλέχθηκε να είναι 100.
2. Ο τρόπος αναπαράστασης κάθε λύσης είναι μια συμβολοσείρα μήκους 16 της μορφής 'BGYRYBGRRGYBGRBG', όπου οι χαρακτήρες B-G-Y-R αντιστοιχούν στα χρώματα Blue-Green-Yellow-Red.
3. Η συνάρτηση καταλληλότητας (fitness function) που επιλέχθηκε, μετράει για κάθε λύση το πλήθος ακμών όπου οι γειτονικοί κόμβοι έχουν το ίδιο χρώμα. Στη συνέχεια, γίνεται κανονικοποίηση των τιμών και τέλος μετατρέπουμε το πρόβλημα ελαχιστοποίησης σε πρόβλημα μεγιστοποίησης.
4. Ο μηχανισμός επιλογής των γονέων προς αναπαραγωγή γίνεται με μερική ανανέωση πληθυσμού (partial population renewal) με ποσοστό 60%, σε συνδυασμό με την μέθοδο «επιλογή τουρνουά (tournament selection)».
5. Η αναπαραγωγή των γονέων γίνεται με διασταύρωση ενός σημείου (one-point crossover).
6. Έπειτα, εφαρμόστηκε μετάλλαξη (mutation) ενός χαρακτήρα στο 10% του πληθυσμού με την μικρότερη τιμή καταλληλότητας (fitness value).
7. Επίσης, εφαρμόστηκε ο μηχανισμός «ελιτισμός (elitism)», περνώντας στην επόμενη γενιά την καλύτερη λύση (μεγαλύτερη τιμή καταλληλότητας).
8. Τέλος, γίνεται τυχαία συμπλήρωση των εναπομεινάντων λύσεων, ώστε ο νέος πληθυσμός να έχει πλήθος 100.

9. Τα βήματα 3 μέχρι 8 επαναλαμβάνονται μέχρι να ικανοποιηθούν οι δύο συνθήκες τερματισμού:
    i. Εύρεση μιας έγκυρης λύσης.
    ii. Τέλος του χρονικού ορίου που επιλέχθηκε να τρέχει το πρόγραμμα μέχρι να βρει μια έγκυρη λύση. Σε περίπτωση που το χρονικό όριο ξεπεραστεί, το πρόγραμμα σταματάει και εμφανίζει την βέλτιστη αλλά όχι ολοκληρωμένη λύση που βρήκε.

Επεξήγηση κώδικα

Το κυρίως πρόγραμμα περιέχει 12 συναρτήσεις:

1. **import_graph:** φορτώνει το λεξικό από το αρχείο ‘graph.json’.
2. **solve_graph_coloring_problem:** αποτελεί την βασική συνάρτηση που καλείται από το κυρίως πρόγραμμα για να αρχίσει η επίλυση του προβλήματος.
3. **generate_initial_solutions:** δημιουργεί το αρχικό πληθυσμό.
4. **fitness_function:** υπολογίζει την τιμή καταλληλότητας της κάθε λύσης.
5. **partial_population_renewal:** υπολογίζει το πλήθος λύσεων που θα χρησιμοποιηθεί στη συνάρτηση «επιλογή τουρνουά», βάσει του δοθέντος ποσοστού μερικής ανανέωσης πληθυσμού.
6. **tournament_selection:** εφαρμόζει την μέθοδο «επιλογή τουρνουά».
7. **one_point_crossover:** υλοποιεί την αναπαραγωγή των γονέων μέσω της διασταύρωση ενός σημείου.
8. **mutation:** υλοποιεί την διαδικασία μετάλλαξης ενός χαρακτήρα σε μερικές λύσεις.
9. **elitism:** υλοποιεί τον μηχανισμό «ελιτισμός».
10. **choose_the_remaining_solutions:** υλοποιεί την τυχαία συμπλήρωση των εναπομεινάντων λύσεων, ώστε ο νέος πληθυσμός να έχει πλήθος 100.
11. **generate_new_population:** δημιουργεί τον πληθυσμό της νέας γενιάς.
12. **visualize_the_solution:** δημιουργεί έναν γράφο με την τελική λύση.

Παραδείγματα εκτέλεσης του προγράμματος