ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ο ΑΛΓΟΡΙΘΜΟΣ ΤΗΣ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μπίκας Ευάγγελος

ΕΠΙΒΛΕΠΩΝ: Δραζιωτης Κωνσταντινος

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στο πλαίσιο αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Υπεύθυνη Δήλωση

(Υπογραφή) ....

Μπίκας Ευάγγελος

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η παρουσίαση και υλοποίηση του αλγόριθμου της αποικίας των μυρμηγκιών. Ο αλγόριθμος αυτός είναι ένας αλγόριθμος βελτιστοποίησης που είναι βασισμένος σε παρατηρήσεις που έγιναν στα μυρμήγκια στην φύση. Πιο συγκεκριμένα, έχει μελετηθεί ο τρόπος με τον οποίο καταφέρνουν να αλληλεπιδρούν μεταξύ τους και το πως βρίσκουν πάντα την βέλτιστη διαδρομή από την φωλιά τους μέχρι την πηγή τροφής.

Αυτό το πετυχαίνουν εκπέμπτοντας χημικές ουσίες που τις χρησιμοποιούν ως σήματα επικοινωνίας, τις λεγόμενες φερομόνες. Ο αλγόριθμος που θα αναλύσουμε είναι μία προσομοίωση αυτής της συμπεριφοράς με χρήση τεχνητών μυρμηγκιών.

Το γεγονός ότι βρίσκει βέλτιστη λύση σε συνδυασμό με το πόσο γρήγορα επιτυγχάνεται και το ότι μπορούν να εφαρμοστούν πολλές παραλλαγές του ανάλογα με το ζητούμενο καθιστά αυτόν τον αλγόριθμο χρήσιμο σε διάφορους τομείς, όπως η επεξεργασία εικόνων και οι τηλεπικοινωνίες. Εντάσεται σε πολλά πεδία και εφαρμογές που έχουν ως στόχο την βελτιστοποίηση, με κλασικό παράδειγμα αυτό του πλανόδιου πωλητή.

Παρακάτω, θα αναλυθεί η θεωρία γράφων και το πως σχετίζεται με αλγόριθμους βελτιστοποίησης, θα αναφερθούν κάποιοι αλγόριθμοι βελτιστοποίησης με έμφαση στον αλγόριθμο της αποικίας των μυρμηγκιών. Θα παρουσιαστεί η δική μου υλοποίηση, θα συγκριθεί με παραλλαγές αυτού και άλλων αλγοριθμων βελτιστοποίησης και θα δωθούν πιθανές εφαρμογές σε προβλήματα πραγματικού κόσμου σε τομείς όπως η επιστήμη των υπολογιστών. Όλοι οι αλγόριθμοι θα υλοποιηθούν σε πψτηον.

Λέξεις Κλειδιά. Προβλήματα Βελτιστοποίησης, Αλγόριθμος Αποικίας Μυρμηγκιών, Θεωρία Γράγων , Python

SUMMARY

The purpose of this thesis is the presentation and implementation of the ant colony algorithm. This algorithm is an optimization algorithm that is based on observations made on ants in nature. More specifically, the way in which they manage to interact with each other and how they always find the optimal route from their nest to the food source has been studied.

They achieve this by emitting chemicals that they use as communication signals, so-called pheromones. The algorithm we will analyze is a simulation of this behavior using artificial ants.

The fact that it finds an optimal solution combined with how fast it is achieved and that many variations of it can be applied depending on the application make this algorithm useful in various fields such as image processing and telecommunications, and span many fields and applications. which aim at optimization, with the classic example of that of the traveling salesman.

Below we will discuss graph theory and how it relates to optimization algorithms, introduce some optimization algorithms with an emphasis on the ant colony algorithm, present my own implementation and compare it to variations of this and other optimization algorithms, and give possible applications in real-world problems in fields such as computer science. All algorithms will be implemented in python.

Key Words. Optimization Problems, Ant Colony Algorithm, Graph Theory, Python

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

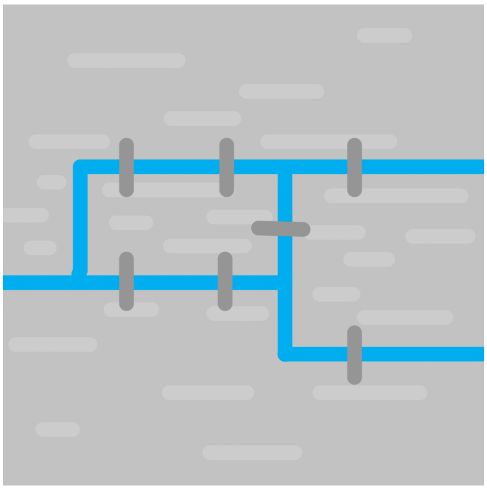
Εισαγωγή

θα γίνει εισαγωγή του θέματος με αναφορά σε προβλήματα πραγματικού κόσμου, όπως του πλανόδιου πωλητή και το πρόβλημα των γεφυρών.

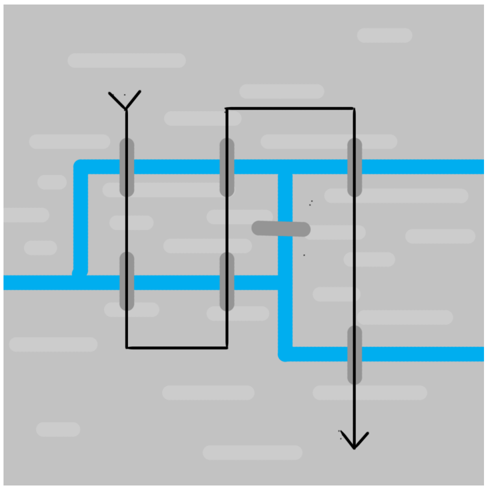
ΚΕΦΑΛΑΙΟ 2

Θεωρία Γράφων

2.1 Ιστορική Αναδρομή

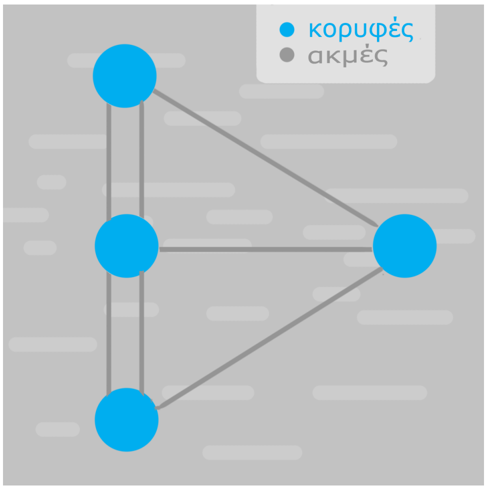


Σχήμα 1: Προσομοίωση Κόνιγκσμπεργκ.

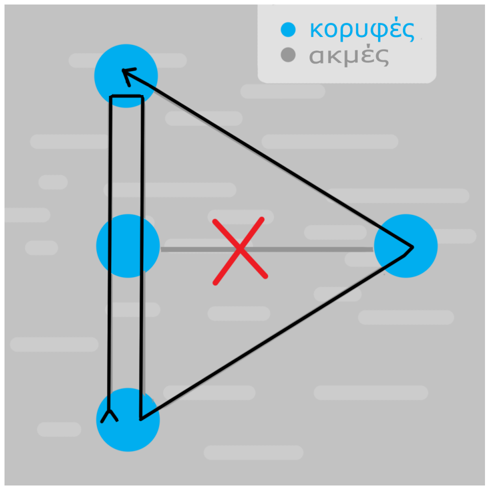


Σχήμα 2: Παράδειγμα διάσχισης γεφυρών

Η ανάπτυξη της θεωρίας γράφων ξεκίνησε τον 18ο αιώνα και πιο συγκεκριμένα το 1736 στην πόλη Königsberg της Πρωσσίας. Σήμερα είναι το Ρωσικό Kaliningrad (μεταξύ Λιθουανίας και Πολωνίας στη Βαλτική) [9]. Η πόλη ήταν χωρισμένη σε 4 τμήματα από τον ποταμό Pregel και χρησιμοποιούνταν 7 γέφυρες για να γίνεται εφικτή η διέλευση των κατοίκων στα διάφορα τμήματά της [Σχήμα 1]. Όταν ο Ελβετός μαθηματικός Λέονχαρντ Όυλερ αναρωτήθηκε αν είναι εφικτό να διασχίσει κάποιος τις γέφυρες της πόλης με βασικό περιορισμό να διασχιστούν όλες οι γέφυρες μόνο μία φορά (Πρόβλημα των γεφυρών του Κόνιγκσμπεργκ) ένας νέος κλάδος των διακριτών μαθηματικών γεννήθηκε, γνωστός και ως θεωρία γράφων. Ο Όυλερ απέδειξε ότι δεν υπάρχει τέτοια διαδρομή μέσω της χρήσης



Σχήμα 3: Κόνιγκσμπεργκ ως γράφος.



Σχήμα 4: Διαδρομή Όυλερ.

γράφων και κατά συνέπεια το πρόβλημα δεν έχει λύση [Σχήμα 2]. Αυτή η απόδειξη απέκτισε αξία όταν ο Ούλερ την εφάρμοσε και σε άλλα προβλήματα γράφων και γενίκευσε την βασική ιδέα.

Μια διαδρομή ονομάζεται διαδρομή Ούλερ όταν μπορούμε να επισκεφτούμε κάθε περιοχή-κορυφή διασχίζοντας την κάθε γέφυρα-ακμή μόνο μία φορά (και ονομάζεται κυκλική αν καταλλήγουμε εκεί που ξεκινήσαμε), αν υπάρχει μια τέτοια διαδρομή σε ένα γράφο τότε αυτός ο γράφος ονομάζεται γράφος Όυλερ. [7] Στο σχήμα [Σχήμα 3] που αντιπροσοπεύει την πόλη του Κόνιγκσμπεργκ σε μορφή γράφου δεν υπάρχει μια τέτοια διαδρομή. Για να γίνει αυτό πρέπει να αφαιρέσουμε μία γέφυρα-ακμή [Σχήμα 4]. Παρατηρήθηκε από τον Όυλερ ότι όλες οι κορυφές πρέπει να έχουν άρτιο βαθμό εκτός από αυτές που ξεκινά και τελειώνει η διαδρομή, εκτός κι αν η διαδρομή είναι κυκλική.

2.2 Εισαγωγή στην Θεωρία Γράφων

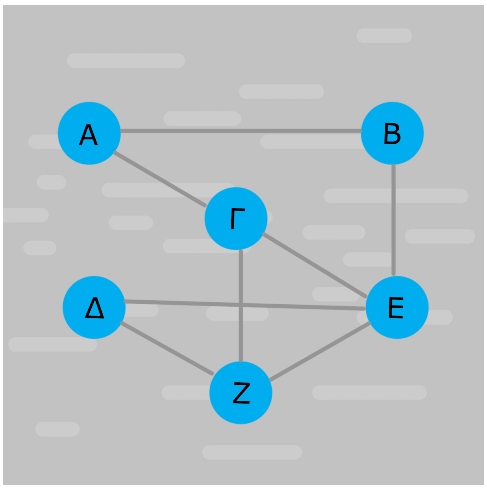
Στην ουσία ένας γράφος είναι διάσπαρτα σημεία (κορυφές) που ενώνονται με γραμμές (ακμές). Γράφους μπορούμε να συνατήσουμε σε διάφορα προβλήματα της καθημερινότητας όπως δίκτυα υποδομών (πχ δίκτυο ύδρευσης), προβλήματα χαρτογράφησης (πλοήγηση), τηλεπικοινωνιών (πχ δορυφόροι), μεταφορών (πχ σιδηρόδρομοι) , και άλλα [9]. Η θεωρία γράφων είναι ένας σημαντικός τομέας των μαθηματικών γιατί πέρα απ᾿ το γεγονός ότι με χρήση αυτών μπορούμε να μοντελοποιήσουμε εύκολα προβλήματα της καθημερινότητας μας σε τομείς που αναφέρθηκαν παραπάνω, μπορούμε επίσης να αναπτύξουμε αλγόριθμους που λύνουν προβλήματα με χρήση γράφων. Παράδειγμα αυτού είναι και ο αλγόριθμος αποικίας των μυρμηγκιών που θα αναλύσουμε σε αυτήν την πτυχιακή εργασία.

2.3 Βασικοί Ορισμοί και έννοιες

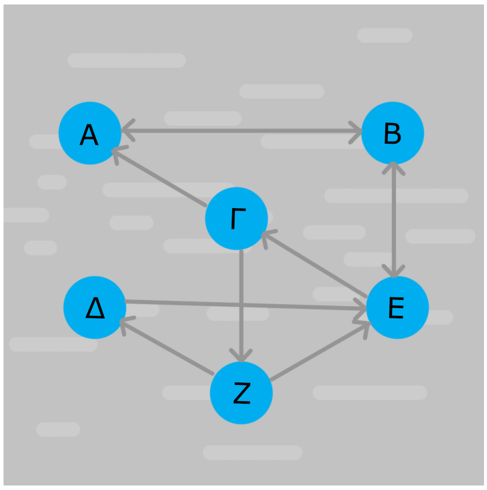
Για να γίνουν κατανοητά όσα θα αναφερθούν στην παρακάτω εργασία είναι απαραίτητο να παρουσιαστεί το θεωρητικό υπόβαθρο πάνω στο οποίο είναι βασισμένοι οι αλγόρθμοι βελτιστοποίησης. Για πιο αναλυτική μελέτη παραπέμπονται οι βιβλιογραφικές αναφορές που χρησιμοποιήθηκαν στο τέλος της εργασίας.

Ένας γράφος είναι μία μαθηματική δομή που ορίζεται με αυστηρό τρόπο μέσω δύο συνόλων: το σύνολο κόμβων (ή κορυφών) και το σύνολο ακμών (ή γραμμών) που συνδέουν ζεύγη κορυφών μεταξύ τους και χρησιμοποιείται για την αναπαράσταση πληροφορίας σχετικά με συνδεσμολογία [7]. Όταν δύο κορυφές είναι συνδεδεμένες, δηλαδή ενώνονται με μία ακμή ονομάζονται γειτονικές (για παραδειγμα στο [Σχήμα 5] η κορυφή Α και η κορυφή Β είναι γειτονικές), αντίστοιχα δύο ακμές που καταλήγουν σε ίδια κορυφή ονομάζονται προσπίπτουσες της κορυφής αυτής. Το πόσες ακμές προσπίπτουν σε μία κορυφή είναι ο βαθμός της κορυφής αυτής. Τάξη ενός γράφου καλούμε το πόσες κορυφές έχει (για παράδειγμα στο [Σχήμα 5] η κορυφή Ε είναι 4ου βαθμού και ο γράφος είναι τάξης 6) . Ένας γράφος μπορεί να είναι είτε κατευθυνόμενος [Σχήμα 6] όταν οι ακμές έχουν μια κατεύθυνση από έναν κόμβο προς έναν άλλο, είτε μη-κατευθυνόμενος [Σχήμα 5] όταν οι ακμές δεν έχουν κατεύθυνση και μπορούν να πηγαίνουν προς οπουδήποτε μεταξύ των κόμβων.

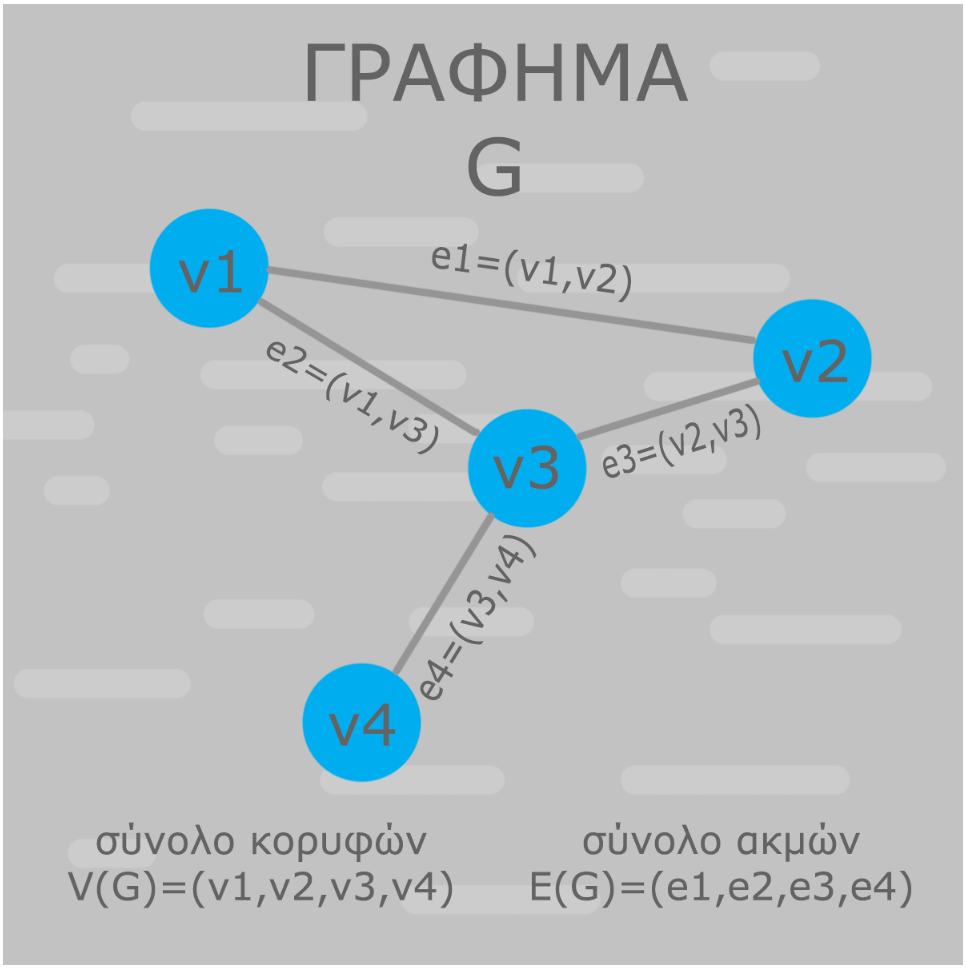
Στις ακμές ενός γράφου μπορούν να επισυναπτούν βάρη, τα οποία αντιπροσωπεύουν το κόστος, την απόσταση ή άλλες χρήσιμες πληροφορίες που συνδέονται με τις σχέσεις μεταξύ των κόμβων. Τα βάρη μπορούν να χρησιμοποιηθούν για την εκτέλεση αλγορίθμων βελτιστοποίησης και την αναζήτηση των βέλτιστων μονοπατιών στον γράφο [7], [6]. Στον αλγόριθμο που θα μελετήσουμε τα βάρη αντιπροσοπεύουν την απόσταση της διαδρομής ή το επίπεδο της φερομόνης (θα αναλυθεί παρακάτω).



Σχήμα 5: Μη κατευθυνόμενος γράφος.



Σχήμα 6: Κατευθυνόμενος γράφος.



Σχήμα 7: Σύνολα γραφήματος.

2.4 Μαθηματικό υπόβαθρο

Ένας γράφος G ορίζεται από δύο σύνολα V και E. Το σύνολο V είναι ένα πεπερασμένο σύνολο (μη άπειρο), που περιέχει ως στοιχεία τις κορυφές του γράφου. Το σύνολο E περιέχει τις ακμές ενός γράφου εκφρασμένες με δισύνολα δύο γειτονικών κορυφών. Έτσι, πεπερασμένος (μη - κατευθυνόμενος) γράφος, λέγεται το διατεταγμένος ζεύγος $G = (V(G), E(G))$ των πεπερασμένων συνόλων $V(G)$, $E(G)$ [7]. Αν πάρουμε ως παράδειγμα το γράφο G στο [Σχήμα 7] παρατήρουμε ότι τα σύνολα $V(G)$ και $E(G)$ έχουν ως εξής:

- $V(G) =[v1, v2, v3, v4]$
- $E(G) =[e1(v1, v2), e2(v1, v3), e3(v2, v3), e4(v3, v4)]$

Επομένως ένας γράφος είναι ένα μαθηματικό αντικείμενο που ορίζεται με αυστηρό τρόπο μέσω δύο συνόλων: το σύνολο κόμβων και το σύνολο ακμών. Το σύνολο των ακμών ενος γράφου μπορεί να είναι κενό, αυτό δεν ισχύει όμως για το σύνολο των κορυφών.

2.5 Αναπαράσταση γράφων

Την κλασσική μορφή αναπαράστασης ενός γράφου την είδαμε ήδη παραπάνω [Σχήμα 3], όμως μια τέτοια αναπαράσταση δεν είναι καθόλου πρακτική σε προγραμματιστικό επίπεδο. Για αυτό αν θέλουμε να αναπαραστίστουμε γράφους σε έναν υπολογιστή χρησιμοποιούμε δομές δεδομένων. Οι δύο πιο βασικοί μέθοδοι αναπαράστασης γράφων σε υπολογιστές είναι οι πίνακες γειτνίασης και οι λίστες γειτνίασης.

Πίνακα γειτνίασης ονομάζουμε ένα πίνακα μεγέθους nxn, όπου n ο αριθμός των κορυφών του γράφου. Κάθε κελί του πίνακα δείχνει την σχέση των αναγραφώμενων κορυφών. Σε ένα μη-κατευθυνόμενος γράφο το κελί (i, j) περιέχει τον αριθμό των ακμών που συνδέουν τον κόμβο i με τον κόμβο j. Σε έναν κατευθυνόμενο γράφο, το κελί (i, j) μπορεί να περιέχει τον αριθμό των ακμών που πηγαίνουν από τον κόμβο i στον κόμβο j. Οι πίνακες γειτνίασης μπορεί να έχουν και βάρη που είναι μια επέκταση του απλού πίνακα γειτνίασης, σε αυτήν την πείπτωση, το έκαστο κελί ενός πίνακα περιέχει έναν αριθμό που ονομάζετε βάρος και υποδηλώνει κάτι ανάλογα με την χρήση του, σε έναν αλγόριθμο βελτιστοποίησης το βάρος μπορεί να υποδηλώνει την απόσταση της μιας κορυφής από την άλλη, την πιθανότητα επιλογής αυτής της διαδρομής, την επιρροή που δέχεται κάποια οντότητα σε επόμενο πιθανό πείραμα ή οποιοδήποτε άλλο κριτήριο που ανταποκρίνεται στον σκοπό του συγκεκριμένου αλγορίθμου.

Ας πάρουμε για παράδειγμα τον γράφο στο [Σχήμα 7], πρόκειται για έναν μη-κατευθυνόμενο γράφο χώρις βάρη με 4 κορυφές, δηλαδή $n = 4$ και με ακμές που φαίνενται στο σύνολο $E(G)$. Επομένως ο πίνακας γειτνίασης του θα διαμορφώνεται έτσι:

$$G_{n,n} = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & v_{1,1} & v_{1,2} & v_{1,3} & v_{1,4} \\ v_2 & v_{2,1} & v_{2,2} & v_{2,3} & v_{2,4} \\ v_3 & v_{3,1} & v_{3,2} & v_{3,3} & v_{3,4} \\ v_4 & v_{4,1} & v_{4,2} & v_{4,3} & v_{4,4} \end{array}$$

που με χρήση 0-1 γίνεται έτσι:

$$G_{4,4} = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 1 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 1 & 0 \end{array}$$

όπου 1 συμβολίζει ότι αυτές οι δύο κορυφές είναι γειτονικές έχοντας ακμή να τις ενώνει, ενώ 0 ότι δεν είναι. Με λίγα λόγια, αν το ζεύγος δύο κορυφών υπάρχει στο σύνολο $E(G)$ τότε το συγκεκριμένο κελί στον πίνακα γειτνίασης θα πάρει την τιμή 1, αλλιώς 0. Σε έναν τέτοιο πίνακα υπάρχει επανάληψη πληροφορίας καθώς το $v(i,j)$ είναι ίδιο με το $v(j,i)$. Σε περίπτωση κατευθυνόμενου γράφου όμως αυτό δε θα συνέβαινε αφού το κελί $v(i,j)$ θα συμβόλιζε αν υπάρχει ακμή από την κορυφή i προς την κορυφή j, ενώ το κελί $v(j,i)$ θα συμβόλιζε αν υπάρχει ακμή από την κορυφή j προς την κορυφή i. Σε περίπτωση ύπαρξης βαρών, τα κελιά με την τιμή 1 στον πίνακα, αντί για 1, θα είχαν την τιμή του αντίστοιχου βαρους.

Λίστα γειτνίασης ονομάζουμε μια αναπαράσταση γράφων όπου για κάθε κορυφή διατηρείται μια λίστα των γειτόνων της. Σε περίπτωση κατευθυνόμενου γράφου μπορεί να υπάρχει ξεχωριστή λίστα για τους εξερχόμενους και τους εισερχόμενους γείτονες. Αυτή η αναπαράσταση αποκτά αξία σε γράφους με αραιή συνδεσιμότητα.

Το πόσο μεγάλη θα είναι η λίστα εξαρτάται από τον αριθμό των κορυφών του γράφου. Κάθε στοιχείο στην λίστα συβολίζει μία ακμή του γράφου.

Σε έναν μη-κατευθυνόμενο γράφο, η λίστα γειτνίασης για κάθε κορυφή περιλαμβάνει τους γείτονές της, δηλαδή τις άλλες κορυφές με τις οποίες συνδέεται με μια ακμή. Αν υπάρχουν n κορυφές στον γράφο, η λίστα γειτνίασης για κάθε κορυφή περιλαμβάνει μια λίστα με το πλήθος των γειτόνων της. Για παράδειγμα στο [Σχήμα 7] που απεικονίζεται ένας μη-κατευθυνόμενος γράφος με 4 κορυφές, άρα $n = 4$ και ακμές που φαίνενται στο σύνολο $E(G)$, αν η αναπαράσταση γινόταν με λίστα γειτνίασης θα ήταν ως εξής:

- $v1 : \{v2, v3\}$
- $v2 : \{v1, v3\}$
- $v3 : \{v1, v2, v4\}$
- $v4 : \{v3\}$

Σε περίπτωση κατευθυνόμενου γράφου σε κάθε κορυφή θα υπάρχαν 2 λίστες, μία που θα εξέφραζε τους προηγούμενες κορυφές και μία που θα εξέφραζε τις ακόλουθες. Για παράδειγμα στο [Σχήμα 6] η κορυφή Γ σε λίστα γειτνίασης θα αναπαριστόταν ως εξής:

- Γ: προγούμενοι:$\{E\}$ ακόλουθοι:$\{A, Z\}$

Σε σχέση με τους πίνακες γειτνίασης οι λίστες γειτνίασης επιτρέπουν την εύκολη πρόσβαση στα δεδομένα του γράφου καθώς και τροποποίηση αυτών, όμως η εισαγωγή και η διαγραφή μιας ακμής είναι πιο χρονοβόρα. Επίσης σε γράφους με λίγες ακμές απαιτούν λίγοτερη μνήμη, ενώ σε πλήρη συνδεδεμένους γράφους περισσότερη.

ΚΕΦΑΛΑΙΟ 3

Αλγόριθμοι Βέλτιστης Διαδρομής (ή βασικοί αλγόριθμοι γράφων)

Ανάλυση και επεξήγηση βασικών αλγορίθμων γράφων, όπως η αναζήτηση πλάτους πρώτης (ΒΦΣ), η αναζήτηση βάθους πρώτης (ΔΦΣ), ο αλγόριθμος της Διϑκστρα για τον ελάχιστο μονοπάτι, ο αλγόριθμος του Κρυσκαλ για το ελάχιστο συνεκτικό δέντρο κ.λπ.

Θα παρουσιαστούν ορισμένοι αλγόριθμοι που ϑα επιλέξω και ϑα γραφούν αναλογα υποπαράγραφοι για τον καθένα

3.1 Εισαγωγή στους Αλγόριθμους Βέλτιστης Διαδρομής

Η εικασία των 4 χρωμάτων πρόβλημα του χρωματισμού χαρτών, αναπαράσταση προσώπων μέσω σημείων και ένωση των σημείων στις περιπτώσεις ύπαρξης προσωπικών σχέσεων, αναπαράσταση μοριακών δομών και αλληλεπιδράσεων τους Παρουσίαση πρακτικών εφαρμογών γράφων, όπως κοινωνικά δίκτυα, μεταφορές, δρομολόγηση, χαρτογραφία, προβλήματα προσβασιμότητας κ.λπ. Επίδραση της δομής του γράφου στην επίλυση προβλημάτων.

3.2 Οι πιο γνωστοί Αλγόριθμοι Βέλτιστης Διαδρομής

3.2.1 Dijkstra

3.2.2 A*

3.2.3 Floyd-Warshall

3.2.4 Αλγόριθμος Αποικίας Μυρμηγκιών

μη ξεχασεις να κανεις αναφορα στη φερομονη.

3.3 Σύγκριση Α.Α.Μ. με υπόλοιπους

ACO [1, 24] is a class of algorithms, whose first member, called Ant System, wasinitially proposed by Colorni, Dorigo and Maniezzo [13, 21, 18]. The main underlying idea, loosely inspired by the behavior of real ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective behavior emerging from the interaction of the different search threads has proved effective in solving combinatorial optimization (CO) problems. Οι μετευρετικοί αλγόριθμοι συνήθως εμπνέονται από φυσικά φαινόμενα, όπως η επιλογή φυσικής εξέλιξης ή η συμπεριφορά των ζώων, και εφαρμόζουν αυτές τις ιδέες στο πρόβλημα που προσπαθούν να λύσουν. Συχνά, οι μετευρετικοί αλγόριθμοι είναι ικανοί να εξερευνήσουν μεγάλους χώρους αναζήτησης και να βρουν καλές λύσεις σε προβλήματα που θεωρούνται δύσκολα για άλλες μέθοδους

2.2 Local search [4]

ΚΕΦΑΛΑΙΟ 4

Α.Α.Μ. σε βάθος

Ο Αλγόριθμος Αποικίας Μυρμηγκιών (Α.Α.Μ.) είναι ένας μετευρετικός αλγόριθμος βελτιστοποίησης που έχει ως στόχο την εύρεσης βέλτιστης διαδρομής σε κάποιο πρόβλημα. Μετευρετικούς ονομάζουμε τους αλγόριθμους που είναι βασισμένοι σε ευφυείς επαναληπτικές τεχνικές και αντί να ακουλουθούν έναν αυστηρό κανόνα εξάγουν γνώση με τρόπους που είναι εμπνευσμένοι από συμπεριφορές στην φύση. [3] Συγκεκριμένα, ο Α.Α.Μ. είναι βασισμένος στην συμπεριφορά τον μυρμηγκιών για την αναζήτηση τροφής. Τα μυρμήγκια έχουν την ικανότητα να βρίσκουν πάντα την βέλτιστη διαδρομή προς την πηγή τροφής όσο δύσκολο κι αν είναι αυτό. Κατά συνέπεια κι ο Α.Α.Μ. είναι ικανός να βρει ικανοποιητικές λύσεις σε περίπλοκα προβλήματα και να εξερευνήσει μεγάλους χώρους αναζήτησης σε μικρό χρονικό διάστημα. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, όταν τα μυρμήγκια κινούνται στο χώρο αφήνουν φερομόνη. Αυτή η ουσία αποτελεί και τρόπο επικοινωνίας μεταξύ των μυρμηγκιών, για εύρεση βέλτιστης διαδρομής προς την τροφή, αφού όσο περισσότερη φερομόνη υπάρχει σε μία διαδρομή, τόσο αυξάναται κι η πιθανότητα ένα επόμενο μυρμήγκι να ακολουθήσει αυτή τη διαδρομή.

4.1 Βασική Θεωρία

Τα τεχνητά μυρμήγκια που χρησιμοποιούνται στον Α.Α.Μ. είναι εμπνευσμένα από την συμπεριφορά των πραγματικών μυρμηγκιών και αποτελούν διαδικασίες κατασκευής στοχαστικών λύσεων που με χρήση πιθανολογικών τεχνικών επιλύουν υπολογιστικά προβλήματα, όπως αυτό της εύρεσης βέλτιστου μονοπατιού μέσω γράφων, προσθέτοντας επαναληπτικά στοιχεία σε επιμέρους λύσεις λαμβάνοντας υπόψη ευρετικές πληροφορίες σχετικά με την επίλυση του προβλήματος και (τεχνητές) διαδρομές φερομόνης που αλλάζουν δυναμικά στο χρόνο εκτέλεσης. [4]

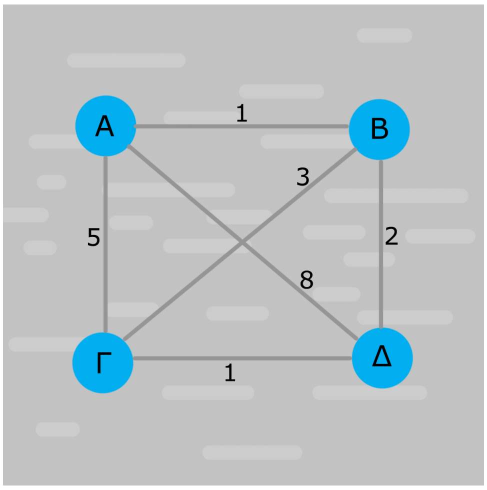


Figure 8: Απόσταση

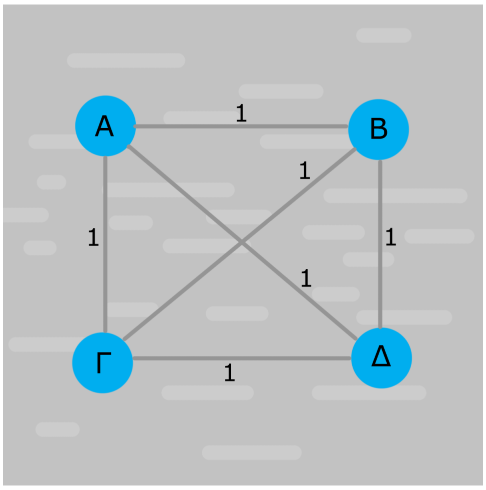


Figure 9: Φερομόνη

4.2 Μαθηματικό υπόβαθρο

4.2.1 Απόσταση

Όπως αναφέρθηκε και στην δεύτερη ενότητα, ο γράφος είναι ένα απαραίτητο κομμάτι για τη μοντελοποίηση προβλημάτων με χρήση του αλγόριθμου αποικίας μυρμηγκιών. Η απόσταση, όπως και η φερομόνη, θα μοντελοποιηθούν με χρήση αναπαράστασης πίνακα. Κάθε ακμή του γράφο έχει ένα κόστος που συμβολίζει την απόσταση ή την γενικότερη ποιότητα της διαδρομής μεταξύ δύο κορυφών. Έστω οι κορυφές Α, Β, Γ, Δ που ενώνονται μεταξύ τους όπως φαίνεται στο [Σχήμα 8] το οποίο αποτυπώνεται σε πίνακα γειτνίασης ως εξής:

$$Distance = \begin{array}{c|cccc} & A & B & \Gamma & \Delta \\ \hline A & 0 & 1 & 8 & 5 \\ B & 1 & 0 & 2 & 3 \\ \Gamma & 8 & 2 & 0 & 1 \\ \Delta & 5 & 3 & 1 & 0 \end{array}$$

4.2.2 Φερομόνη

Τα μυρμήγκια αφήνουν φερομόνη από όπου περνούν ανάλογα από την ποιότητα της λύσης που βρήκαν ώστε να γνωρίζουν τα επόμενα. Τα μονοπάτια φερομόνης χρησιμεύουν ως αριθμητικές πληροφορίες που χρησιμοποιούν τα μυρμήγκια για να κατασκευάσουν πιθανές λύσεις στο εκάστοτε πρόβλημα και τα οποία μεταβάλονται κατά την εκτέλεση του αλγορίθμου με στόχο της εύρεση του βέλτιστου μονοπατιού. [4] Σε μονοπάτια που το μυρμήγκι θα έχει επιστρέψει σε σύντομο χρονικό διάστημα συσσωρεύεται περισσότερη φερομόνη από άλλα μονοπάτια. Υπάρχουν μοντέλα που παράγωγουν περισσότερη φερομόνη ανάλογα με την ποιότητα αυτής της διαδρομής (για παράδειγμα την απόσταση, την ποσότητα της τροφής, και άλλα)

Το μαθηματικό μοντέλο αναπαράστασης της φερομόνης που εξάγει το κ-οστό μυρμήγκι στην ακμή που ενώνει τις κορυφές i και j (δηλαδή η ποσότητα της φερομόνης που παράγει) είναι αντιστρόφος ανάλογη με την απόσταση και προκύπτει από τον τύπο:

$$\Delta\tau_{i,j}^{k} = \frac{1}{L_k} \tag{1}$$

Όπου:

- L_k: Η ποιότητα της διαδρομής που βρήκε το μυρμήγκι

Έστω ότι για το πρώτο μυρμήγκι είναι παντού 1 [Σχήμα 9]. Με αποτέλεσμα να είναι τυχαία η επιλογή διαδρομής. Στο παράδειγμα μας προκύπτει ο πίνακας:

$$Pheromone = \begin{array}{c|cccc} & A & B & \Gamma & \Delta \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ \Gamma & 1 & 1 & 0 & 1 \\ \Delta & 1 & 1 & 1 & 0 \end{array}$$

Οπότε ως έδώ έχουμε ένα χάρτη με τις πιθανά μονοπάτια στον πίνακα distance και το αντίστοιχο κόστος της κάθε διαδρομής και έναν ακόμα με την "επιθυμία" του μυρμηγκιού να επιλέγει το κάθε μονοπάτι στον πίνακα pheromone.

Για να υπολογίσουμε την ποσότητα φερομόνης από μια κορυφή σε μία άλλη (χωρίς εξασθένση- θα αναλυθεί παρακάτω) υπολογίζουμε το άθροισμα της φερομόνης που εξήγαγαν τα μυρμήγκια m που πέρασαν από την κορυφή i στην j. Δηλαδή:

$$\tau_{i,j}^{k} = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \tag{2}$$

4.2.3 Επιλογή Διαδρομής

Στο παράδειγμα μας ας υποθέσουμε ότι ένα μυρμήγκι ξεκινάει στην κορυφή A. Οι πιθανές του επιλογές όπως φαίνεται και στο [Σχήμα 10] είναι οι B, Γ, Δ. Ποιά είναι όμως η βέλτιστη; Με το μάτι σε ένα τόσο απλό πρόβλημα είναι εύκολο να αντιληφθούμε ποιά διαδρομή πρέπει να ακολουθήσει το μυρμήγκι, όμως αυτό δεν είναι εφικτό σε πιο περίπλοκα προβλήματα με μεγάλο αριθμό κορυφών. Έστω ότι ο αριθμός των κορυφών είναι n τότε επιλέγοντας μία κορυφη ως αρχική ο αριθμός των πιθανών κορυφών γίνεται n-1. Αφού το μυρμήγκι επιλέξει μία από αυτή μετά θα αφαιρεθεί από τις πιθανές αφού επισκέφθηκε και θα γίνουν n-2. Έτσι

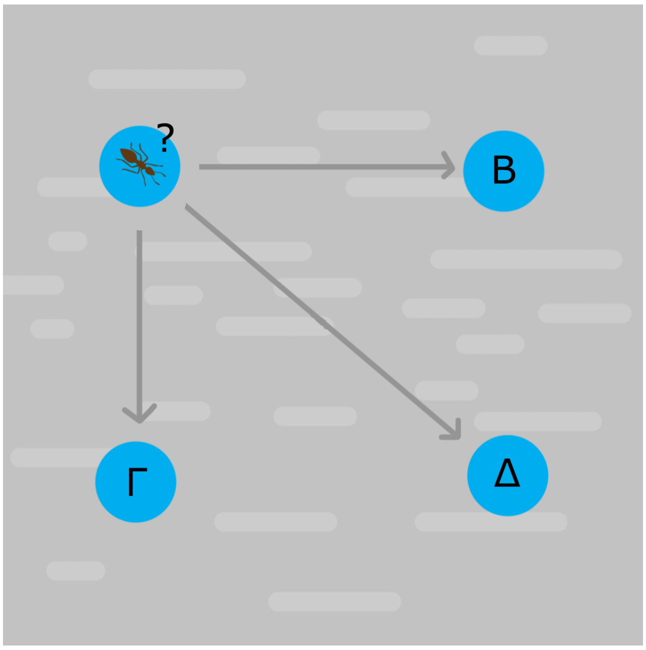


Figure 10: Πιθανές επιλογές

θα επιλέγει μονοπάτια μεχρι να μην μείνει κανένα διαθέσιμο και να επιστρέψει στο αρχικό. Άρα ο αριθμός των πιθανών επιλογών είναι (n-1)(n-2)(n-3)...3*2*1 = (n-1)!. Όμως δεδομένου ότι διαδρομές όπως Α->Β->Γ->Δ->Α είναι ίδια με την Α->Δ->Γ->Β->Α. Οπότε αφού πρόκειται για συμμετρικό πρόβλημα ο τύπος αριθμός των πιθανών μονοπατιών γίνεται (n-1)!/2 αφαιρώντας τις επαναλαμβανόμενες λύσεις όπως και στο πρόβλημα του πλανόδιου πωλητή. Η πιθανότητα το κ-οστό μυρμήγκι να επιλέξει μια διαδρομή συμβολίζεται ως: $P_{i,j}^{k}$ και δίνεται από τον τύπο: [5]

$$P_{i,j}^{k} = \frac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum_m (\tau_{i,m})^{\alpha}(\eta_{i,m})^{\beta}} \tag{3}$$

Όπου:

- $\tau_{i,j}$: το επίπεδο φερομόνης μεταξύ των κορυφών i και j
- $\eta_{i,j}$: είναι η ποιότητα της διαδρομής
- m: η υπόλοιπες διαδρομές που μπορούσε να επιλέξει το μυρμήγκι
- α, β: σταθερές που επιλέγουμε ανάλογα από την επιρροή που θέλουμε να έχει το τ και το η στην διαδικασία επιλογής. (Για παράδειγμα αν θέλουμε να επιλέξουμε μια διαδρομή βασισμένη αποκλιστικά και μόνο στο επίπεδο της φερομόνης τότε αφαιρούμε από την εξίσοση το $\eta_{i,j}$ θέτοντας το β=0).

Το γινόμενο των $\tau_{i,j} * \eta_{i,j}$ μας δίνει την "επιθυμία" του μυρμηγκιού να επιλέξει το μονοπάτι i,j. Στις περισσότερες βιβλιογραφίες αυτή η πιθανότητα αναφέρετε έτσι με διαφορετικούς συμβολισμούς στην καθεμία.

Στο παράδειγμα μας, αφού υπολογίσουμε την επιθυμία του μυρμηγκιού να επιλέξει την κάθε διαδρομή έχουμε:

- A->B $= \tau_{A,B}\eta_{A,B} = 1 * \frac{1}{1} = 1$
- A->Γ $= \tau_{A,\Gamma}\eta_{A,\Gamma} = 1 * \frac{1}{5} = 0.2$
- A->Δ $= \tau_{A,\Delta}\eta_{A,\Delta} = 1 * \frac{1}{8} = 0.125$

Όπου οι αντίστοιχες πιθανότητες γίνονται:

- $P_{A,B} = \frac{1}{1+0.125+0.2} = \frac{1}{1.325} = 0.76$
- $P_{A,\Gamma} = \frac{0.2}{1.325} = 0.15$
- $P_{A,\Delta} = \frac{0.125}{1.325} = 0.09$

Βλέπουμε ότι πιο πιθανό είναι το μυρμήγκι να επιλέξει την διαδρομή ΑΒ όμως υπάρχει πιθανότητα να επιλέξει και τις υπόλοιπες, για να προσομοιώσουμε την τυχαία επιλογή μονοπατιού με βάση την πιθανότητα χρησιμοποιούμε την τεχνική roulette wheel [1], όπου υπολογίζουμε το άθροισμα συσσωρευμένο, αφού πρόκειτε για πιθανότητα το άθροισμα τους είναι 1, γίνεται η τυχαία επιλογή ενός αριθμού από το 0 έως το 1 και βρίσκουμε το σύνολο στο οποίο αντιστοιχεί αυτός ο αριθμός. Στο παράδειγμά μας από 0 έως 0,76 αντιστοιχεί στην διαδρομή ΑΒ, από το 0,77 εως το 0,91 αντιστοιχεί στην διαδρομή ΑΓ και απο το 0,92 έως το 1 στην διαδρομή ΑΔ. Όποτε αν για παράδειγμα επιλεγόταν ο αριθμός 0,56 θα πηγαίναμε στην διαδρομή ΑΒ.

4.2.4 Εξασθένιση Φερομόνης

(4.2.2 αναφέρθηκε)

4.3 Υλοποίησή μου Αλγορίθμου σε python

(4.2.3 τυχαία επιλογή αριθμού από τις πιθανότητες)

4.4 Εφαρμογές Αλγόριθμου

Θα δωθούν προβλήματα πραγματικού κόσμου που λύνει αυτός ο αλγόριθμος

4.5 Βελτιστοποίηση Αλγόριθμου

References

[1] Dorota Lipowska Adam Lipowski. Roulette-wheel selection via stochastic acceptance, 2011.

[2] Christian Blum. Ant colony optimization: Introduction and recent trends, 2005.

[3] Vittorio Maniezzo. Ant colony optimization: an overview, 2002.

[4] Thomas Stutzle Marco Dorigo. The ant colony optimization metaheuristic: Algorithms, applications, and advances, 2002. Technical Report IRIDIA-2000-32.

[5] Chirag Chandrashekar-Pradeep Krishnadoss-Vijayakumar Kedalu Poornachary-Balasundaram Ananthakrishnan-Kumar Rangasamy. Hybrid weighted ant colony optimization algorithm for task scheduling in cloud computing, 2023.

[6] Γκέρτσης Γεώργιος. Θεωρία Γράφων Και Εύρεση Βέλτιστης Διαδρομή, 2023.

[7] ΝΤΕΝΙΣΙΩΤΗΣ ΙΩΑΝΝΗΣ. Θεωρία Γράφων και Εφαρμογές της, 2021.

[8] Γεωργιάδης Ιωάννης. Θεωρία Γράφων και αλγόριθμοι εύρεσης βέλτιστης διαδρομής σε δίκτυα, 2017.

[9] Ιωάννης Μανωλόπουλος. Αλγοριθμική Θεωρία Γράφων. Εισαγωγή στη Θεωρία Γράφων, 2014. Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

Παράρτημα

Εγκατάσταση και εκτέλεση Anaconda διανομής και jupyter notebook

Εκτέλεση του κώδικα