

Σχολή Επιστήμης της Πληροφορίας & Πληροφορικής

Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ)

«Ψηφιακές Εφαρμογές και Καινοτομία»

«Πρακτική υλοποίηση προγραμμάτων για την επίλυση προβλημάτων βελτιστοποίησης μέσω του κβαντικού υπολογιστή D-Wave»

Ιάσων – Ιωάννης Παπαναγιώτου

Τριμελής Επιτροπή

Επόπτης Καθηγητής:
Θεόδωρος Ανδρόνικος

Μέλη:
Δημήτριος Τσουμάκος
Μιχαήλ Στεφανιδάκης

Κέρκυρα, Φεβρουάριος 2020

Περίληψη

Η παρούσα εργασία ασχολείται με την πρακτική υλοποίηση προγραμμάτων τα οποία μπορούν να εκτελεστούν στον κβαντικό υπολογιστή D-Wave και επιλύουν προβλήματα βελτιστοποίησης.

Ειδικότερα, το πρώτο κεφάλαιο αποτελεί μια εισαγωγή στο κβαντικό υπολογισμό, στις βασικές αρχές και λειτουργίες αυτού, στη διαφορά μεταξύ qubits και bits και στα πλεονεκτήματα των κβαντικών υπολογιστών.

Στο δεύτερο κεφάλαιο παρουσιάζεται ο κβαντικός υπολογιστής D-Wave και αναλύονται τα δομικά του στοιχεία από πλευράς υλικού και λογισμικού, όπως το περιβάλλον λειτουργίας και τα διαθέσιμα εργαλεία για την ανάπτυξη εφαρμογών.

Στο τρίτο κεφάλαιο παρέχεται ένας συνοπτικός οδηγός για την εγκατάσταση των απαραίτητων λογισμικών που απαιτούνται για την εκτέλεση προγραμμάτων στον κβαντικό υπολογιστή D-Wave. Στη συνέχεια εξετάζονται τέσσερα NP-πλήρη προβλήματα βελτιστοποίησης και αναπτύσσονται προγράμματα σε γλώσσα Python για την επίλυσή τους με τη χρήση των βιβλιοθηκών του κβαντικού υπολογιστή.

Λέξεις – Κλειδιά

Κβαντικός υπολογιστής, Βελτιστοποίηση, NP-πλήρη προβλήματα, Python, D-Wave, TSP

Abstract

This diploma thesis deals with the practical implementation of programs that can be executed on the D-Wave quantum computer and solve optimization problems.

In particular, the first chapter provides an introduction to quantum computing, its basic principles and functions as well as the difference between qubits and bits and the advantages of quantum computing.

The second chapter presents the D-Wave quantum computer and analyzes its hardware and software components, such as its operating environment and the available tools for application development.

The third chapter provides a brief guide for installing the software needed to run programs on the D-Wave quantum. Additionally, four NP-complete optimization problems are examined and Python programs are developed to solve them using quantum computer.

Keywords

Quantum computing, Optimization, NP-complete problems, Python, D-Wave, TSP

Περιεχόμενα

Περίληψη iii
Abstract iv
Περιεχόμενα v
Κατάλογος Εικόνων / Σχημάτων vi
Κατάλογος Πινάκων viii
Συντομογραφίες & Ακρωνύμια ix
1. Εισαγωγή στο κβαντικό υπολογισμό 1
1.1 Βασικές αρχές 1
1.2 Σύγκριση bits και qubits 3
1.3 Λειτουργίες 5
1.4 Η εξέλιξη και τα πλεονεκτήματα των κβαντικών υπολογιστών 5
2. Ο κβαντικός υπολογιστής D-Wave 7
2.1 Περιγραφή 7
2.2 Ιστορική αναδρομή 9
2.3 Εφαρμογές 12
2.4 Η τοπολογία Chimera 14
2.5 Η τοπολογία Pegasus 16
2.6 Περιβάλλον προγραμματισμού D-Wave 18
3. Επίλυση προβλημάτων στον D-Wave 22
3.1 Εγκατάσταση Ocean SDK 22
3.2 Το πρόβλημα του μέγιστου συνόλου ανεξαρτησίας 26
3.3 Το πρόβλημα του χρωματικού αριθμού γραφήματος 32
3.4 Το πρόβλημα του χρωματισμού χάρτη 37
3.5 Το πρόβλημα του πλανόδιου πωλητή (TSP) 40
Βιβλιογραφία 46

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 2-1 Περιβάλλον λειτουργίας D-Wave QPU 8
Εικόνα 2-2 D-Wave QPU 9
Εικόνα 3-1 Επαλήθευση εγκατάστασης Ocean SDK 24
Εικόνα 3-2 LEAP: API URL, Authenitcation Token....25
Εικόνα 3-3 Επιτυχής αποστολή προβλήματος στον D-Wave....26
Εικόνα 3-4 ΜΣΑ για το γράφημα G_1....30
Εικόνα 3-5 ΜΣΑ για το γράφημα G_2....31
Εικόνα 3-6 Χρωματισμός γραφήματος G_3....35
Εικόνα 3-7 Χρωματισμός γραφήματος G_4....36
Εικόνα 3-8 Χρωματισμός γράφου «Ελλάδα» 39
Εικόνα 3-9 Χρωματισμός χάρτη Ελλάδας 40
Εικόνα 3-10 TSP για το γράφημα G_5....44
Εικόνα 3-11 TSP για το γράφημα G_6....45

Σχήμα 2-1 Η εξέλιξη των κβαντικών υπολογιστών D-Wave 2004-2017 12
Σχήμα 2-2 C3 γράφημα Chimera (3x3): Τα qubits οργανώνονται σε 9 κυψελοειδείς μονάδες 15
Σχήμα 2-3 Διάταξη των qubits σε μια κυψελοειδή μονάδα 16
Σχήμα 2-4 Επεξεργαστής Pegasus(0) 17
Σχήμα 2-5 Περιβάλλον προγραμματισμού D-Wave 18
Σχήμα 2-6 Αρχιτεκτονική λογισμικού D-Wave 19
Σχήμα 3-1 Γράφοι G_1, G_2 29
Σχήμα 3-2 Γράφοι G_3, G_4 34
Σχήμα 3-3 Οι διοικητικές περιφέρειες της Ελλάδας ως γράφος 38
Σχήμα 3-4 Γράφοι G_5, G_6 43

Κατάλογος Πινάκων

Πίνακας 2-1 Η εξέλιξη των κβαντικών υπολογιστών D-Wave 2011-2020 11
Πίνακας 2-2 Δειγματολήπτες Ocean 21

Συντομογραφίες & Ακρωνύμια

Qubit	Quantum bit
API	Application Programming Interface
SAPI	Solver API
SDK	Software Development Kit

1. Εισαγωγή στο κβαντικό υπολογισμό

Ο κβαντικός υπολογιστής είναι μια υπολογιστική συσκευή που εκμεταλλεύεται χαρακτηριστικές ιδιότητες της κβαντομηχανικής, όπως την αρχή της υπέρθεσης και της διεμπλοκής καταστάσεων, προκειμένου να επεξεργασθεί δεδομένα και να εκτελέσει υπολογισμούς.

Ο κβαντικός υπολογισμός συνιστά ένα σύγχρονο ακαδημαϊκό πεδίο που εξετάζει τη λειτουργία των κβαντικών υπολογιστών και την διατύπωση κατάλληλων αλγορίθμων από τη σκοπιά της θεωρητικής πληροφορικής, ενώ η σχετική πρακτική τεχνολογία είναι ακόμα σε πρώιμα στάδια ανάπτυξης (Βικιπαίδεια, 2019).

1.1 Βασικές αρχές

Η βασική αρχή της κβαντικής υπολογιστικής επιστήμης είναι το γεγονός ότι οι κβαντομηχανικές ιδιότητες της ύλης μπορούν να χρησιμοποιηθούν για την αναπαράσταση και τη δόμηση δεδομένων, καθώς και το γεγονός ότι μπορούν να επινοηθούν και να κατασκευαστούν μηχανισμοί στηριγμένοι στην κβαντομηχανική για την επεξεργασία αυτών των δεδομένων.

Σε έναν κλασικό συμβατικό ψηφιακό υπολογιστή (κατά κανόνα ηλεκτρονικό), στοιχειώδης μονάδα πληροφορίας είναι το bit, που μπορεί να πάρει μόνο δύο τιμές: «0» και «1». Σε έναν κβαντικό υπολογιστή μονάδα πληροφορίας είναι το κβαντικό bit (quantum bit), το οποίο για συντομία γράφεται ως qubit. Όπως και στην κλασική θεωρία των πληροφοριών, απαιτούνται κβαντικά συστήματα δύο καταστάσεων (qubits) για την κωδικοποίηση και την επεξεργασία των πληροφοριών (Bertoni, Bordone, Brunetti, Jacoboni, & Reggiani, 2000, p. 5912), (Kalai, 2016, p. 509).

Το qubit είναι ένα κβαντικό σύστημα δύο καταστάσεων, που συμβολίζονται με $|0\rangle$ και $|1\rangle$. Η μνήμη ενός κλασικού ψηφιακού υπολογιστή αποτελείται από bits τα οποία μπορούν να αναπαραστήσουν την τιμή 1 ή 0. Ένα qubit μπορεί να αναπαραστήσει την τιμή 1, 0 ή οποιαδήποτε υπέρθεση αυτών των δύο (Bennett & DiVincenzo, 2000, p. 248).

Υπάρχουν αρκετά κβαντικά συστήματα δύο καταστάσεων τα οποία μπορούν να χρησιμοποιηθούν ως qubits.

Για παράδειγμα η κατάσταση περιστροφής (spin) ενός σωματιδίου με spin $^1/_2$ μπορεί να θεωρηθεί ως qubit, όπου η κατάσταση spin $+^1/_2$ (πάνω) αντιστοιχεί στη βασική κατάσταση $|0\rangle$, ενώ η κατάσταση spin $-^1/_2$ (κάτω) στη βασική κατάσταση $|1\rangle$:

$$\left|+\frac{1}{2}\right\rangle \rightarrow |0\rangle \text{ και } \left|-\frac{1}{2}\right\rangle \rightarrow |1\rangle$$

Η διεύθυνση ομοίως πόλωσης ενός φωτονίου μπορεί να αναπαραστήσει ένα qubit, όπου η οριζόντια πόλωση αντιστοιχεί στην κατάσταση $|0\rangle$ και η κάθετη στην κατάσταση $|1\rangle$. Το qubit μπορεί επίσης να παρασταθεί και από τον τρόπο ταλάντωσης δύο ιόντων που βρίσκονται μέσα σε μία παγίδα ιόντων.

Το qubit μπορεί να βρεθεί σε οποιαδήποτε υπέρθεση των δύο βασικών καταστάσεων:

$$|q\rangle = a|0\rangle + b|1\rangle \text{ όπου } |a|^2 + |b|^2 = 1.$$

Οι δύο βασικές καταστάσεις του qubit μπορούν να γραφούν ως πίνακες:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ και } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

Το ίδιο και η υπέρθεση καταστάσεων:

$$|q\rangle = \mathbf{a}|0\rangle + \mathbf{b}|1\rangle = \mathbf{a}\begin{bmatrix}1\\0\end{bmatrix} + \mathbf{b}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix}$$

Οι αριθμοί a και b ονομάζονται πλάτη πιθανότητας, είναι μιγαδικοί αριθμοί και το διάνυσμα κατάστασης $|q\rangle$ του qubit είναι ένα διάνυσμα στο χώρο Hilbert που έχει δύο διαστάσεις. Οι βασικές καταστάσεις ενός qubit είναι ορθογώνιες μεταξύ τους (Καραφυλλίδης, 2015, σσ. 28-29):

$$\langle 0|0\rangle = 1 \qquad \langle 0|1\rangle = 0$$
$$\langle 1|0\rangle = 0 \qquad \langle 1|1\rangle = 1$$

Δύο (2) qubits μπορούν να αναπαραστήσουν οποιαδήποτε υπέρθεση τεσσάρων (4) δυνατών καταστάσεων και τρία (3) qubits οποιαδήποτε υπέρθεση οκτώ (8) καταστάσεων. Γενικά ένας κβαντικός υπολογιστής με n qubits μπορεί να βρίσκεται σε αυθαίρετη υπέρθεση των έως 2^n δυνατών καταστάσεων ταυτόχρονα, ενώ ένας κλασικός υπολογιστής μπορεί να βρίσκεται μόνο σε μιαν από αυτές τις καταστάσεις κάθε στιγμή.

Ο κβαντικός υπολογιστής λειτουργεί θέτοντας τα qubits σε μιαν ελεγχόμενη αρχική κατάσταση που αναπαριστά το αρχικό πρόβλημα και χειρίζεται τα qubits χρησιμοποιώντας

λογικές κβαντικές πύλες. Η αλληλουχία των πυλών που χρησιμοποιούνται ονομάζεται κβαντικός αλγόριθμος.

Στην πραγματικότητα οποιοδήποτε σύστημα έχει μια ποσότητα *A* που μπορεί να παρατηρηθεί, -η οποία διατηρείται με την εξέλιξη του χρόνου και είναι τέτοια, ώστε η *A* να έχει τουλάχιστον δύο διακριτές και επαρκώς κατανεμημένες διαδοχικές ιδιοτιμές-, είναι κατάλληλο για να υλοποιήσει ένα qubit. Αυτό συμβαίνει επειδή ένα τέτοιο σύστημα μπορεί να χαρτογραφηθεί πάνω σε ένα αποτελεσματικό σύστημα με περιστροφή $^1/_2$ (spin-$^1/_2$) (DiVincenzo, 2000, σ. 2).

1.2 Σύγκριση bits και qubits

Ένας υπολογιστής με έναν αριθμό qubits είναι θεμελιωδώς διαφορετικός από έναν κλασικό υπολογιστή με τον ίδιο αριθμό bits. Σε έναν κλασικό υπολογιστή για να αναπαρασταθεί π.χ. η κατάσταση ενός συστήματος με n-qubits χρειάζεται να αποθηκευτούν 2n μιγαδικοί συντελεστές. Το γεγονός αυτό δείχνει ότι τα qubits μπορούν να αποθηκεύσουν εκθετικά περισσότερη πληροφορία από τα κλασικά bits. Όμως δεν πρέπει να παραβλεφθεί ότι τα qubits είναι μόνον μια πιθανολογική υπέρθεση όλων των πιθανών καταστάσεών τους. Αυτό σημαίνει ότι όταν μετρήσουμε την τελική κατάσταση των qubits θα βρίσκονται μόνο σε έναν από τους πιθανούς σχηματισμούς που βρίσκονταν πριν τη μέτρηση. Είναι λάθος να υποτεθεί, ότι τα qubits βρίσκονταν σε μία συγκεκριμένη κατάσταση πριν την μέτρηση, εφόσον το γεγονός ότι ήταν σε μια υπέρθεση καταστάσεων πριν την μέτρηση επηρεάζει τα πιθανά αποτελέσματα του υπολογισμού.

Έστω λ.χ. έναν κλασικός υπολογιστής που λειτουργεί πάνω σε έναν καταχωρητή με 3 bits. Η κατάσταση του υπολογιστή σε οποιαδήποτε στιγμή είναι μια πιθανότητα κατανεμημένη σε 2^3=8 διαφορετικές 3-bites ακολουθίες: 000, 001, 010, 011, 100, 101, 110, 111. Όταν είναι ντετερμινιστικός υπολογιστής, τότε θα βρίσκεται ακριβώς σε μιαν από αυτές τις καταστάσεις με πιθανότητα 1. Ωστόσο αν είναι πιθανολογικός υπολογιστής, υπάρχει πιθανότητα να βρίσκεται σε μια πληθώρα καταστάσεων. Μπορούμε να περιγράψουμε αυτή την πιθανολογική κατάσταση με οκτώ (8) μη αρνητικούς αριθμούς A, B, C, D, E, F, G, H (όπου A = η πιθανότητα ο υπολογιστής να βρίσκεται στην κατάσταση 000, B = η πιθανότητα να βρίσκεται στην κατάσταση 001, κλπ.). Το άθροισμα όλων αυτών των πιθανοτήτων είναι 1.

Η κατάσταση ενός 3-bit-ου κβαντικού υπολογιστή περιγράφεται από ένα διάνυσμα με οκτώ διαστάσεις (*a, b, c, d, e, f, g, h*), που ονομάζεται ket. Ωστόσο, αντί το άθροισμά τους να είναι 1, το άθροισμα των τετραγώνων των συντελεστών, $|a|^2+|b|^2+...+|h|^2$ πρέπει να είναι 1. Επίσης οι συντελεστές μπορούν να έχουν σύνθετες τιμές. Το απόλυτο τετράγωνο των συντελεστών υποδηλώνει το πλάτος πιθανότητας των δοθεισών καταστάσεων. Η φάση μεταξύ οποιωνδήποτε δύο συντελεστών (καταστάσεις) αναπαριστά μια βαρυσήμαντη παράμετρο, η οποία αποτελεί μια θεμελιώδη διαφορά μεταξύ των κβαντικών υπολογιστών και των πιθανολογικών κλασικών υπολογιστών.

Αν μετρήσουμε τα τρία (3) qubits, θα δούμε μια ακολουθία τριών (3) bits. Η πιθανότητα μέτρησης μιας δοθείσας ακολουθίας ισούται με το τετράγωνο αυτής της ακολουθίας συντελεστή (Π.χ. η πιθανότητα μέτρησης 000=$|a|^2$, η πιθανότητα μέτρησης 001=$|b|^2$ κλπ.). Έτσι, η μέτρηση μιας κβαντικής κατάστασης που περιγράφεται από σύνθετους συντελεστές (a, b, ..., h) δίνει την κλασική κατανομή πιθανότητας ($|a|^2$, $|b|^2$, ..., $|h|^2$), οπότε η κβαντική κατάσταση "καταρρέει" σε μια κλασική κατάσταση σαν αποτέλεσμα πραγματοποίησης μιας μέτρησης.

Να σημειωθεί ότι ένα διάνυσμα οκτώ διαστάσεων μπορεί να καθοριστεί με διάφορους τρόπους ανάλογα με την βάση που επιλέγεται το διάστημα. Η βάση των ακολουθιών από bits (λ.χ. 000, 001, ..., 111) είναι γνωστή ως βάση υπολογισμού. Άλλες πιθανές βάσεις είναι μοναδιαία ορθογώνια διανύσματα και τα ιδιοδιανύσματα του τελεστή Pauli-x. Το νεύμα ket συνήθως χρησιμοποιείται για να κάνει την επιλογή σαφέστατη.

Για παράδειγμα, η κατάσταση (*a, b, c, d, e, f, g, h*) στην βάση υπολογισμού μπορεί να γραφεί ως εξής:

$$a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle,$$
$$\text{όπου λ.χ. } |010\rangle = (0,0,1,0,0,0,0,0)$$

Η βάση υπολογισμού ενός qubit δύο διαστάσεων είναι: $|0\rangle = (1,0)$ και $|1\rangle = (0,1)$.

Χρησιμοποιώντας τα ιδιοδιανύσματα του τελεστή Pauli-x ένα qubit είναι (Βικιπαίδεια, 2019):

$$\left|+\right\rangle = \frac{1}{\sqrt{2}}(1,1) \text{ και } \left|-\right\rangle = \frac{1}{\sqrt{2}}(1,-1)$$

1.3 Λειτουργίες

Παρότι μια κλασική κατάσταση τριών-bits και μια κβαντική κατάσταση τριών-bits είναι διανύσματα οκτώ διαστάσεων, τα διαχειριζόμαστε διαφορετικά για κλασικούς ή κβαντικούς υπολογισμούς. Και στις δύο περιπτώσεις το σύστημα πρέπει να αρχικοποιηθεί, για παράδειγμα στην μηδενική ακολουθία $|000\rangle$ που αντιστοιχεί στο διάνυσμα (0, 0, 1, 0, 0, 0, 0, 0). Στον κλασικό τυχαιοποιημένο υπολογισμό, το σύστημα εξελίσσεται σύμφωνα με την εφαρμογή στοχαστικών μητρών, οι οποίες διαφυλάσσουν ότι οι πιθανότητες δίνουν άθροισμα το 1. Αντίθετα στους κβαντικούς υπολογισμούς, επιτρέπονται λειτουργίες ενιαίας μήτρας, οι οποίες είναι αποδοτικές περιστροφές. Συνεπώς, αφού οι περιστροφές μπορούν να αναιρεθούν, κάνοντας αντίστροφες περιστροφές, οι κβαντικοί υπολογισμοί είναι αναστρέψιμοι.

Τελικά, κατά τον τερματισμό του αλγορίθμου, το αποτέλεσμα πρέπει να διαβαστεί. Στην περίπτωση του κλασικού υπολογιστή έχουμε δείγμα από την κατανομή πιθανοτήτων πάνω σε έναν καταχωρητή τριών bits για να πάρει μιαν οριστική ακολουθία τριών bits, ας πούμε 000. Στην κβαντική μηχανική μετράμε τη κατάσταση τριών qubits, η οποία είναι ισοδύναμη με την κατάρρευση της κβαντικής κατάστασης σε κανονική κατανομή (με τους συντελεστές στην κλασική κατάσταση να είναι τετραγωνικά μεγέθη των συντελεστών για την κβαντική κατάσταση, όπως αναφέρθηκε παραπάνω), ακολουθούμενη από δειγματοληψία από αυτήν την κατανομή. Να σημειωθεί ότι αυτό καταστρέφει την κανονική κβαντική κατάσταση. Πολλοί αλγόριθμοι δίνουν την σωστή απάντηση με κάποια πιθανότητα. Ωστόσο, από την επαναλαμβανόμενη αρχικοποίηση, το να τρέχουμε και να κάνουμε μετρήσεις στον κβαντικό υπολογιστή, αυξάνει την πιθανότητα να πάρουμε την σωστή απάντηση (Βικιπαίδεια, 2019).

1.4 Η εξέλιξη και τα πλεονεκτήματα των κβαντικών υπολογιστών

Παρά την απίστευτη ισχύ των σημερινών υπερυπολογιστών, πολύπλοκα προβλήματα πληροφορικής δεν μπορούν να αντιμετωπιστούν από τα συμβατικά συστήματα. Η τεράστια ανάπτυξη δεδομένων και η ανάγκη να κατανοήσουμε καλύτερα τα πάντα, -από το σύμπαν μέχρι το DNA-, μας οδηγεί στην αναζήτηση νέων εργαλείων που μπορούν να βοηθήσουν στην εύρεση των απαντήσεων. Ο κβαντικός υπολογισμός είναι το επόμενο σύνορο,

παρέχοντας μια εντελώς νέα προσέγγιση στην επίλυση των δυσκολότερων προβλημάτων (D-Wave Systems Inc., 2018, p. 1).

Αν και οι κβαντικοί υπολογιστές δεν βρίσκονται ακόμα σε πλήρη εμπορική εφαρμογή, τα αποτελέσματα των πρώτων αυτών συσκευών με σχετικά περιορισμένο πλήθος από qubits είναι ενθαρρυντικά. Μεγάλης κλίμακας κβαντικοί υπολογιστές αναμένεται να λύνουν προβλήματα πολύ ταχύτερα από τους κλασικούς υπολογιστές, χρησιμοποιώντας τους καλύτερους μέχρι τώρα γνωστούς αλγόριθμους, όπως η παραγοντοποίηση μεγάλων αριθμών χρησιμοποιώντας τον αλγόριθμο του Shor ή η προσομοίωση μεγάλων συστημάτων (Nielsen & Chuang, 2010, p. 172).

Αν δοθούν αρκετοί υπολογιστικοί πόροι σε έναν κλασικό υπολογιστή, μπορεί να προσομοιώσει οποιοδήποτε κβαντικόν αλγόριθμο. Ωστόσο η υπολογιστική ισχύς 500 qubits, για παράδειγμα, θα ήταν ήδη πολύ μεγάλη για να αναπαρασταθεί σε έναν κλασικό υπολογιστή, γιατί θα χρειαζόταν να αποθηκευτούν 2^{500} τιμές (ένα terabyte πληροφορίας μπορεί να αποθηκεύσει 2^{43} διακριτές τιμές).

Ο κβαντικός υπολογισμός είναι ένας σχετικά νέος τομέας, ιδιαίτερα σύνθετος και ακολουθούνται διαφορετικές προσεγγίσεις σε όλο τον κόσμο. Σήμερα υπάρχουν δύο κύριες αρχιτεκτονικές για τους κβαντικούς υπολογιστές: το μοντέλο πύλης και η κβαντική ανόπτηση (quantum annealing).

Στο κβαντικό υπολογιστικό μοντέλο πύλης, ο στόχος είναι να ελέγχει και να χειρίζεται την εξέλιξη των κβαντικών καταστάσεων με την πάροδο του χρόνου - μια δύσκολη πρόκληση, ειδικά σε μεγάλες κλίμακες, επειδή τα κβαντικά συστήματα είναι εξαιρετικά ευαίσθητα.

Η κβαντική ανόπτηση εκμεταλλεύεται τη φυσική εξέλιξη των κβαντικών καταστάσεων (D-Wave Systems Inc., 2018, p. 1).

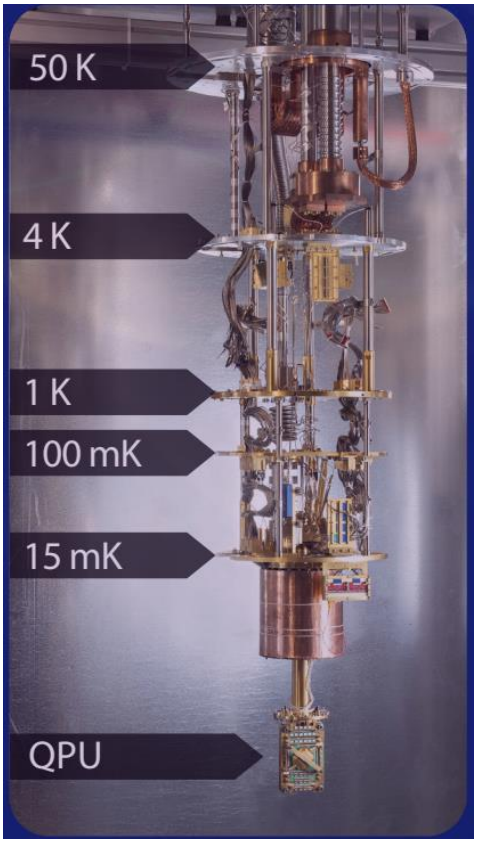
2. Ο κβαντικός υπολογιστής D-Wave

2.1 Περιγραφή

Η εταιρεία κβαντικών υπολογιστών D-Wave εφαρμόζει την κβαντική ανόπτηση σαν ένα μοναδικό κβαντικό αλγόριθμο για να δημιουργήσει τις κβαντικές μονάδες επεξεργασίας (QPUs-Quantum Processing Units), που αναφέρονται επίσης και ως κβαντικά τσιπς, περιέχουν έναν αριθμό διασυνδεδεμένων qubits και συνιστούν το βασικό συστατικό ενός πλήρους κβαντικού υπολογιστή.

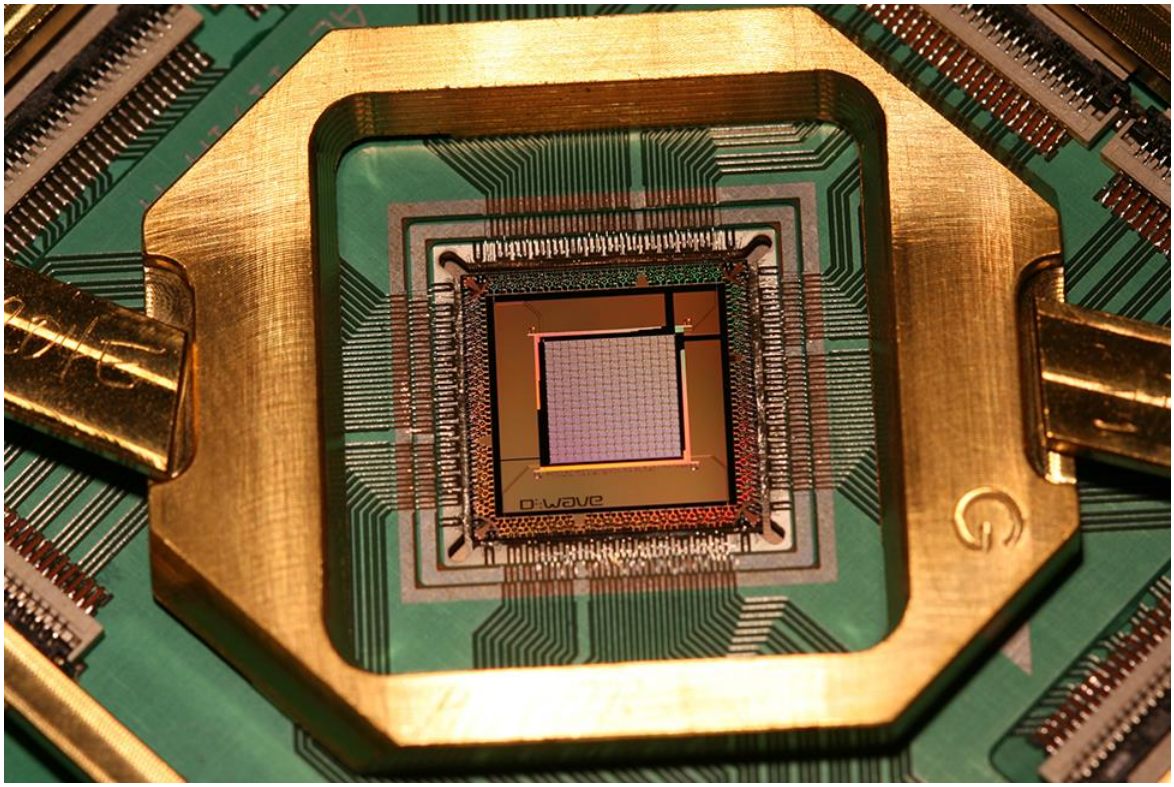
Το σύστημα D-Wave περιέχει μια QPU που πρέπει να διατηρείται σε θερμοκρασία κοντά στο απόλυτο μηδέν και να απομονώνεται από το περιβάλλον, ώστε να συμπεριφέρεται σύμφωνα με τους κανόνες της κβαντομηχανικής. Το σύστημα επιτυγχάνει αυτές τις απαιτήσεις ως εξής:

- Κρυογονικές θερμοκρασίες που εξασφαλίζονται με τη χρήση ψυγείου κρυογονικής αραίωσης κλειστού βρόγχου. Η QPU λειτουργεί σε θερμοκρασίες κάτω από 15 mK. Ξεκινώντας από θερμοκρασία δωματίου, η θερμοκρασία μειώνεται σε κάθε επίπεδο μέχρι να φτάσει σχεδόν το απόλυτο μηδέν στο σημείο όπου βρίσκεται η QPU (Εικόνα 2-1).
- Κάλυψη από ηλεκτρομαγνητικές παρεμβολές που επιτυγχάνεται με τη χρήση θωρακισμένου περιβλήματος ραδιοσυχνοτήτων (RF) και υποσυστήματος μαγνητικής θωράκισης.



Εικόνα 2-1 Περιβάλλον λειτουργίας D-Wave QPU

Η QPU της D-Wave είναι ένα πλέγμα από μικροσκοπικούς μεταλλικούς βρόγχους, καθένας από τους οποίους είναι ένα qubit ή ένας ζεύκτης (coupler). Κάτω από θερμοκρασίες 9,2 kelvin, αυτοί οι βρόγχοι γίνονται υπεραγωγοί και εμφανίζουν κβαντομηχανικά αποτελέσματα.



Εικόνα 2-2 D-Wave QPU

Η D-Wave 2000Q QPU έχει μέχρι 2048 qubits και 6016 ζεύκτες. Για να φτάσει σε αυτήν την κλίμακα χρησιμοποιεί 128,000 κόμβους Josephson, γεγονός που καθιστά την D-Wave 2000Q QPU το πιο πολύπλοκο υπεραγώγιμο ολοκληρωμένο κύκλωμα.

2.2 Ιστορική αναδρομή

Ο πρώτος κβαντικός υπολογιστής που κατασκευάστηκε από την εταιρία D-Wave Systems είχε την ονομασία Orion. Στις 13 Φεβρουαρίου του 2007 η D-Wave παρουσίασε το σύστημα του υπολογιστή Orion, «τρέχοντας» σε αυτό τρεις διαφορετικές εφαρμογές στο Μουσείο Ιστορίας Υπολογιστών της Mountain View στην Καλιφόρνια. Ο Orion έχει ως βάση του τα 16 qubits και είναι κατασκευασμένος από ένα υλικό, που είναι μέτριος αγωγός θερμότητας και ηλεκτρισμού και παρ' όλα αυτά είναι ένα αρκετά σκληρό μέταλλο: το νιόβιο. Ωστόσο, τα qubits για να διατηρούν την κβαντική τους κατάσταση, πρέπει να είναι παγωμένα. Για τον λόγο αυτό ο Orion έχει θερμοκρασία που αγγίζει το απόλυτο μηδέν.

Στις 11 Μαΐου του 2011 η D-Wave Systems ανακοίνωσε τον D-Wave One, έναν ολοκληρωμένο κβαντικό υπολογιστή, ο οποίος τρέχει σε έναν επεξεργαστή 128-qubits. Ο

επεξεργαστής που χρησιμοποιείται στον D-Wave One με την κωδική ονομασία "Rainier", εκτελεί μία και μόνη μαθηματική πράξη, την διακριτή βελτιστοποίηση. Ο Rainier χρησιμοποιεί κβαντική ανόπτηση για την επίλυση προβλημάτων βελτιστοποίησης και είναι το πρώτο εμπορικά διαθέσιμο κβαντικό σύστημα υπολογιστών στον κόσμο.

Στις αρχές του 2012 η D-Wave Systems παρουσίασε έναν 512-qubits κβαντικό υπολογιστή, τον D-Wave Two με την κωδική ονομασία Vesuvius, που είναι ο δεύτερος εμπορικά διαθέσιμος κβαντικός υπολογιστής και ο διάδοχος του πρώτου εμπορικά διαθέσιμου υπολογιστή D-Wave One. Οι υπολογιστές αυτοί έχουν σχεδιαστεί για την εκτέλεση αδιαβατικών κβαντικών υπολογισμών. Από τον Μάρτιο του 2014 συζητείται έντονα αν μία μεγάλης κλίμακας εμπλοκή πραγματοποιείται στον D-Wave Two και εάν οι μελλοντικές γενιές των υπολογιστών D-Wave θα έχουν κάποιο πλεονέκτημα έναντι των κλασσικών υπολογιστών. Ο D-Wave Two με επεξεργαστή 512-qubits συνιστά μια τεράστια βελτίωση σε σχέση με τον D-Wave One, ο οποίος διαθέτει 128-qubits QPU. Αν κάθε qubit ήταν σε άμεση επικοινωνία με κάθε άλλο qubit, τότε μια μηχανή 512-qubits θα ήταν πάνω από 100 τάξεις μεγέθους πιο δύσκολη για την προσομοίωση από μια μηχανή 128-qubits. Ωστόσο, επειδή κάθε qubit στην αρχιτεκτονική D-Wave επικοινωνεί απευθείας μόνον με έξι άλλα qubits, η D-Wave εκτιμά ότι ο υπολογιστής D-Wave Two είναι περίπου 300.000 φορές πιο ισχυρός από ότι ο D-Wave One. Τον Μάιο του 2013 η Catherine McGeoch, μία σύμβουλος της D-Wave, δημοσίευσε την πρώτη σύγκριση της κβαντικής τεχνολογίας με ένα κλασσικό top-end desktop γενικού σκοπού (CPLEX), που τρέχει έναν αλγόριθμο βελτιστοποίησης. Χρησιμοποιώντας μια διαμόρφωση με 439 qubits, το σύστημα “έτρεξε” 3.600 φορές πιο γρήγορα από τον CPLEX, με τον καλύτερο αλγόριθμο για την συμβατική μηχανή να επιλύει προβλήματα με 100 ή περισσότερες μεταβλητές σε μισή ώρα, εν αντιθέσει με τον κβαντικό υπολογιστή που χρειάστηκε μισό δευτερόλεπτο. Βέβαια η σύγκριση δεν είναι "αρκετά δίκαιη”, επειδή ο υπολογιστής γενικού σκοπού θα εκτελέσει τις διεργασίες πάντα πιο αργά από ότι μια συσκευή αφιερωμένη στην επίλυση ενός συγκεκριμένου προβλήματος (Χριστοπούλος, 2015).

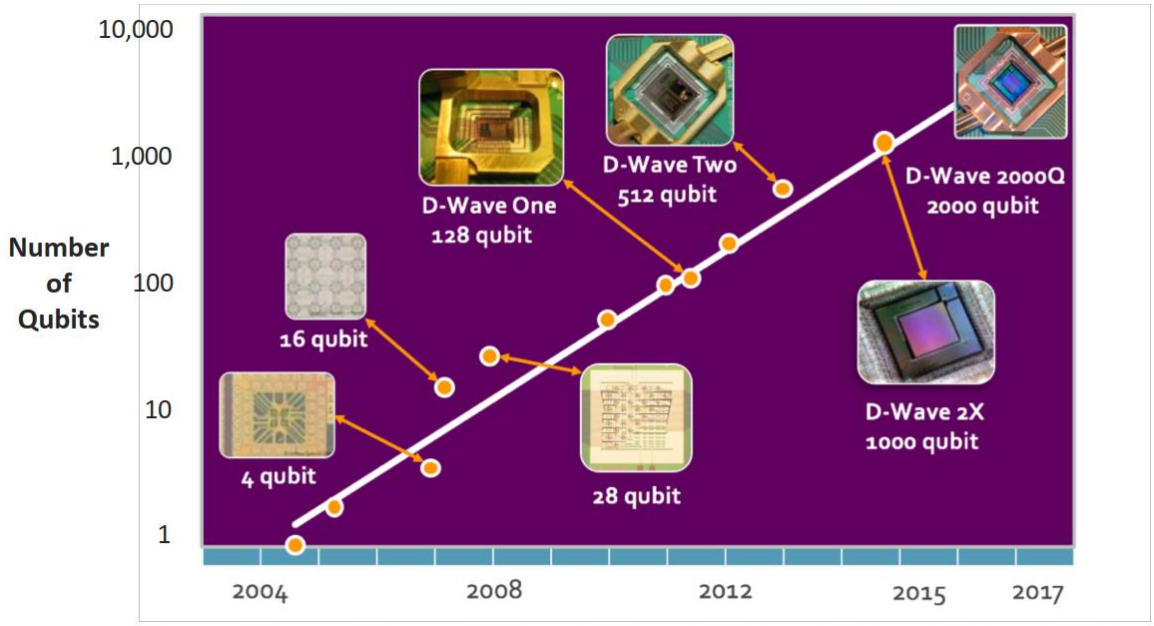
Τον Αύγουστο του 2015 η εταιρεία ανακοινώνει τη διάθεση του D-Wave 2X ο οποίος διαθέτει 1000 qubits σε αρχιτεκτονική γράφου Chimera. Ο συνολικός αριθμός των qubits που είναι διαθέσιμα προς χρήση είναι περίπου λιγότερα από 1152, γιατί ο ακριβής αριθμός των qubits που παράγονται θα διαφέρει ανάλογα με τον κάθε επεξεργαστή που κατασκευάστηκε.

Τον Ιανουάριο του 2017 ανακοινώθηκε ο D-Wave 2000Q με 2048 qubits, ενώ η εταιρεία τον Φεβρουάριο του 2019 ανακοίνωσε την επόμενη γενιά της πλατφόρμας κβαντικών επεξεργαστών με την ονομασία "Pegasus" των 15 συνδέσεων ανά qubit αντί των 6. Η διάθεση του νέου συστήματος αναμένεται στα μέσα του 2020.

Η διαχρονική εξέλιξη των κβαντικών υπολογιστών της εταιρείας D-Wave συνοψίζεται αναλυτικά στον παρακάτω πίνακα και συνοπτικά στο επόμενο σχήμα (Pakin, Dahl, & Ewald, 2017, σ. 17):

	D-Wave One	**D-Wave Two**	**D-Wave 2X**	**D-Wave 2000Q**	**Advantage**
Ημερομηνία διάθεσης	Μάιος 2011	Μάιος 2013	Αύγουστος 2015	Ιανουάριος 2017	2020
Κωδική ονομασία	Rainier	Vesuvius	W1K	W2K	TBA
Qubits	128	512	1152	2048	> 5000
Θερμοκρασία λειτουργίας (K)	-	0,02	0,015	0,015	TBA
Κατανάλωση ενέργειας (kW)	-	15,5	25	25	TBA
Αγοραστές	Lockheed Martin	Lockheed Martin, Google, NASA, USRA	Lockheed Martin, Google, NASA, USRA, Los Alamos National Laboratory	Temporal Defense Systems, Google. NASA. USRA	Los Alamos National Laboratory

Πίνακας 2-1 Η εξέλιξη των κβαντικών υπολογιστών D-Wave 2011-2020



Σχήμα 2-1 Η εξέλιξη των κβαντικών υπολογιστών D-Wave 2004-2017

2.3 Εφαρμογές

Οι κβαντικοί υπολογιστές D-Wave μπορούν να χρησιμοποιηθούν για την επίλυση πολλών και δύσκολων προβλημάτων στους τομείς της βελτιστοποίησης, της μηχανικής μάθησης και της ασφάλειας στον κυβερνοχώρο (D-Wave Systems Inc.).

Με 2000 qubit και νέα χαρακτηριστικά ελέγχου, ο κβαντικός υπολογιστής D-Wave 2000Q μπορεί να λύσει μεγαλύτερα προβλήματα από ό,τι ήταν στο παρελθόν δυνατό και με καλύτερες επιδόσεις. Μια αναπτυσσόμενη κοινότητα προγραμματιστών χρησιμοποιούν τις δυνατότητες των συστημάτων D-Wave για ποικίλες εφαρμογές σε διάφορους τομείς όπως:

Μηχανικής μάθησης:

- Ανίχνευση στατιστικών ανωμαλιών
- Αναγνώριση εικόνων και προτύπων
- Εκπαίδευση νευρωνικών δικτυών
- Επικύρωση λογισμικού
- Ταξινόμηση μη δομημένων δεδομένων
- Διάγνωση λαθών κυκλωμάτων
- Ανάλυση και μοντελοποίηση εκλογών

Βελτιστοποίησης:

- Κυκλοφοριακής ροής
- Διαφήμισης στο διαδίκτυο
- Τηλεπικοινωνιακών δικτύων
- Ηλεκτρονικό εμπόριο

Οικονομία:

- Ανίχνευση διακυμάνσεων αγορών
- Ανάπτυξη εμπορικών στρατηγικών
- Βελτιστοποίηση εμπορικών συναλλαγών
- Βελτιστοποίηση χαρτοφυλακίου

Επιστήμη υλικών:

- Προσομοίωση κβαντικών συστημάτων
- Προτυποποίηση υλικών

Ασφάλεια:

- Ανίχνευση κακόβουλου λογισμικού και επιθέσεων στο δίκτυο
- Προγραμματισμός πόρων
- Ανάλυση γραφημάτων
- Αναγνώριση προτύπων και ανίχνευση ανωμαλιών
- Προσομοίωση κβαντικών συστημάτων
- Προτυποποίηση υλικών

Υγειονομική περίθαλψη:

- Εντοπισμός περιπτώσεων απάτης
- Δημιουργία στοχευμένων θεραπειών ενάντια στο καρκίνο
- Βελτιστοποίηση των ακτινοθεραπειών
- Μοντελοποίηση πρωτεϊνών

2.4 Η τοπολογία Chimera

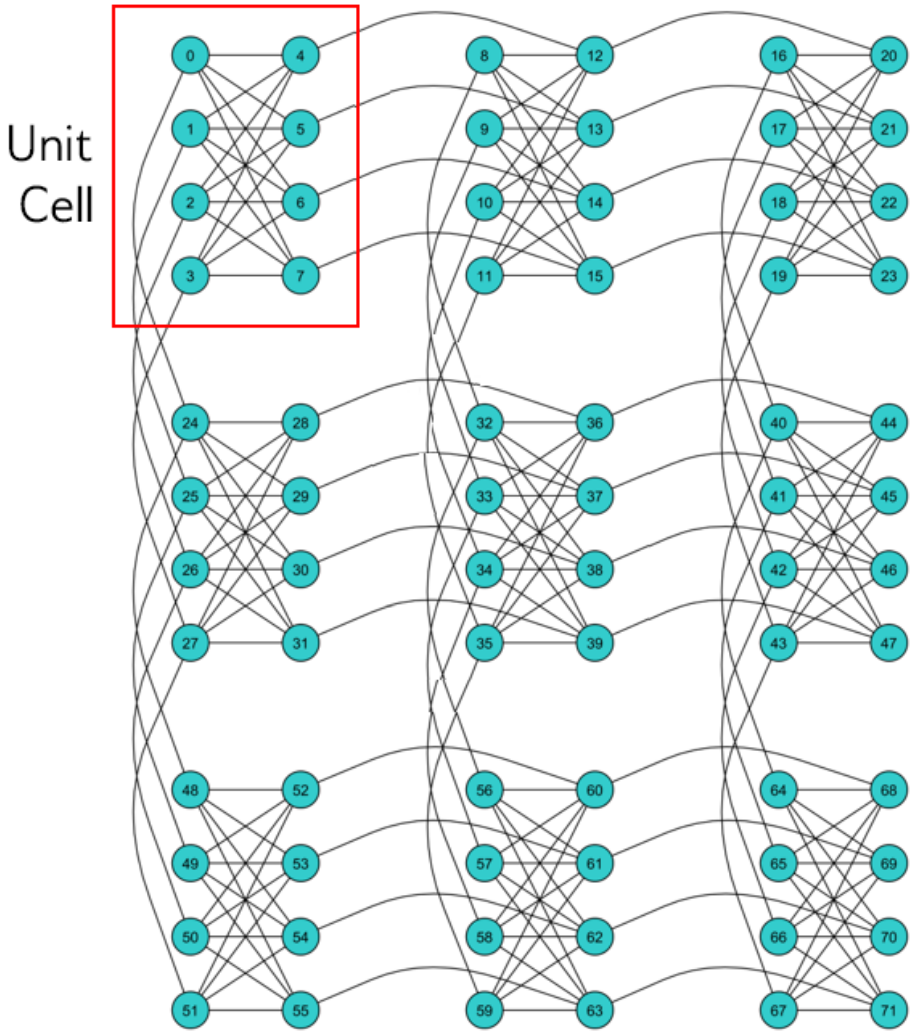
Η D-Wave QPU συνιστά ένα πλέγμα διασυνδεδεμένων qubits. Ενώ μερικά qubits συνδέονται με άλλα μέσω ζευκτών, η D-Wave QPU δεν είναι πλήρως συνδεδεμένη. Αντί τούτου τα qubits διασυνδέονται σε μια αρχιτεκτονική που είναι γνωστή ως «Χίμαιρα» (“Chimera”).

Η κατανόηση της διάταξης του D-Wave QPU είναι ζωτικής σημασίας για τη μετάφραση μιας αντικειμενικής συνάρτησης σε μορφή που να μπορεί να λύσει ένα σύστημα D-Wave. Γνωρίζουμε ότι οι δυαδικές αντικειμενικές συναρτήσεις μπορούν να αναπαρασταθούν ως γράφοι, αλλά προτού μπορέσουμε να τις στείλουμε προς επίλυση από την QPU, πρέπει πρώτα να κατανοήσουμε καλύτερα την αρχιτεκτονική της.

Η αρχιτεκτονική Chimera αποτελείται από σετ συνδεδεμένων κυψελοειδών μονάδων (unit cells) που περιλαμβάνουν τέσσερα οριζόντια qubits συνδεδεμένα σε τέσσερα κάθετα qubits μέσω ζευκτών. Αυτές οι κυψελοειδείς μονάδες (unit cells) είναι οργανωμένες σε οριζόντια και κάθετη διάταξη, δημιουργώντας ένα πλέγμα από αραιά συνδεδεμένα qubits.

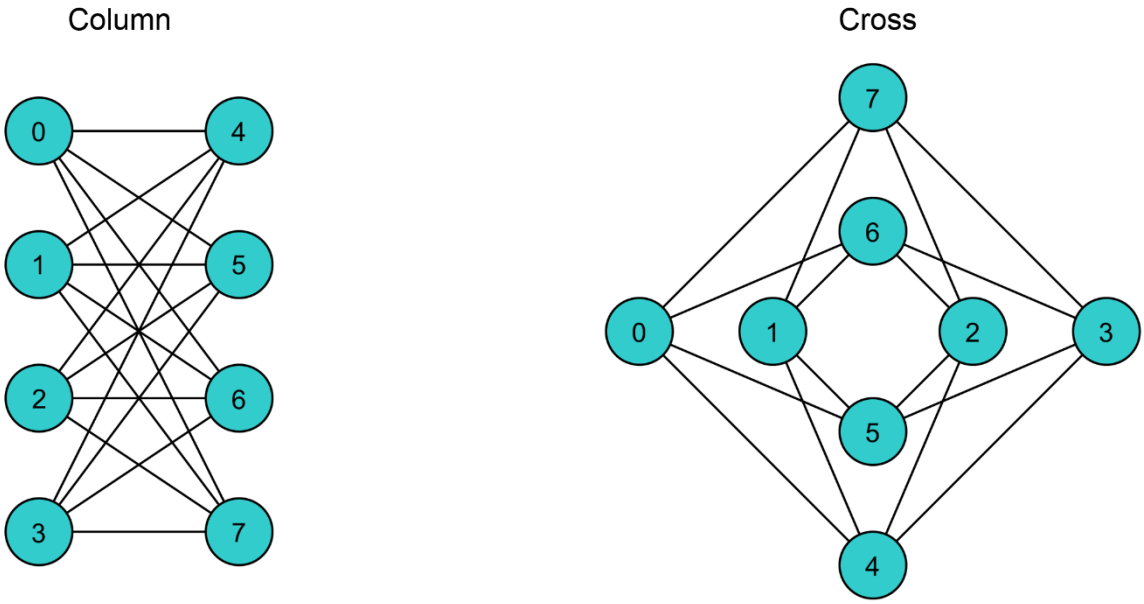
Ο συμβολισμός CN αναφέρεται σε ένα γράφημα Chimera που αποτελείται από ένα πλέγμα NxN κυψελοειδών μονάδων (Σχήμα 2-2).

Έτσι, η D-Wave 2000Q QPU υποστηρίζει γράφημα C16: τα 2048 qubits αντιστοιχίζονται σε μήτρα 16x16 κυψελοειδών μονάδων των 8 qubits.



Σχήμα 2-2 C3 γράφημα Chimera (3x3): Τα qubits οργανώνονται σε 9 κυψελοειδείς μονάδες

Μια κυψελοειδής μονάδα μπορεί να οργανωθεί σε διάταξη σταυρού (cross) ή διάταξη στήλης (column). Σε κάθε διάταξη υπάρχουν δύο σύνολα των τεσσάρων qubits. Κάθε qubit του ενός συνόλου είναι συνδεδεμένο με όλα τα qubits του άλλου συνόλου, αλλά όχι με τα qubits του συνόλου που ανήκει. Συνεπώς, σε κάθε κυψελοειδή μονάδα τα qubits έχουν διμερή συνδεσιμότητα (bipartite connectivity), όπως φαίνεται και στο Σχήμα 2-3.



Σχήμα 2-3 Διάταξη των qubits σε μια κυψελοειδή μονάδα

Οι κορυφές και οι ακμές του γράφου που αναπαριστούν μια αντικειμενική συνάρτηση «μεταφράζονται» σε qubits και ζεύκτες στη Chimera. Κάθε λογικό qubit, στο γράφημα της αντικειμενικής συνάρτησης, μπορεί να αναπαρασταθεί με ένα ή περισσότερα φυσικά qubits. Η διαδικασία της αντιστοίχισης των λογικών qubits με τα φυσικά ονομάζεται minor embedding (D-Wave Systems Inc., 2018, σσ. 17-19).

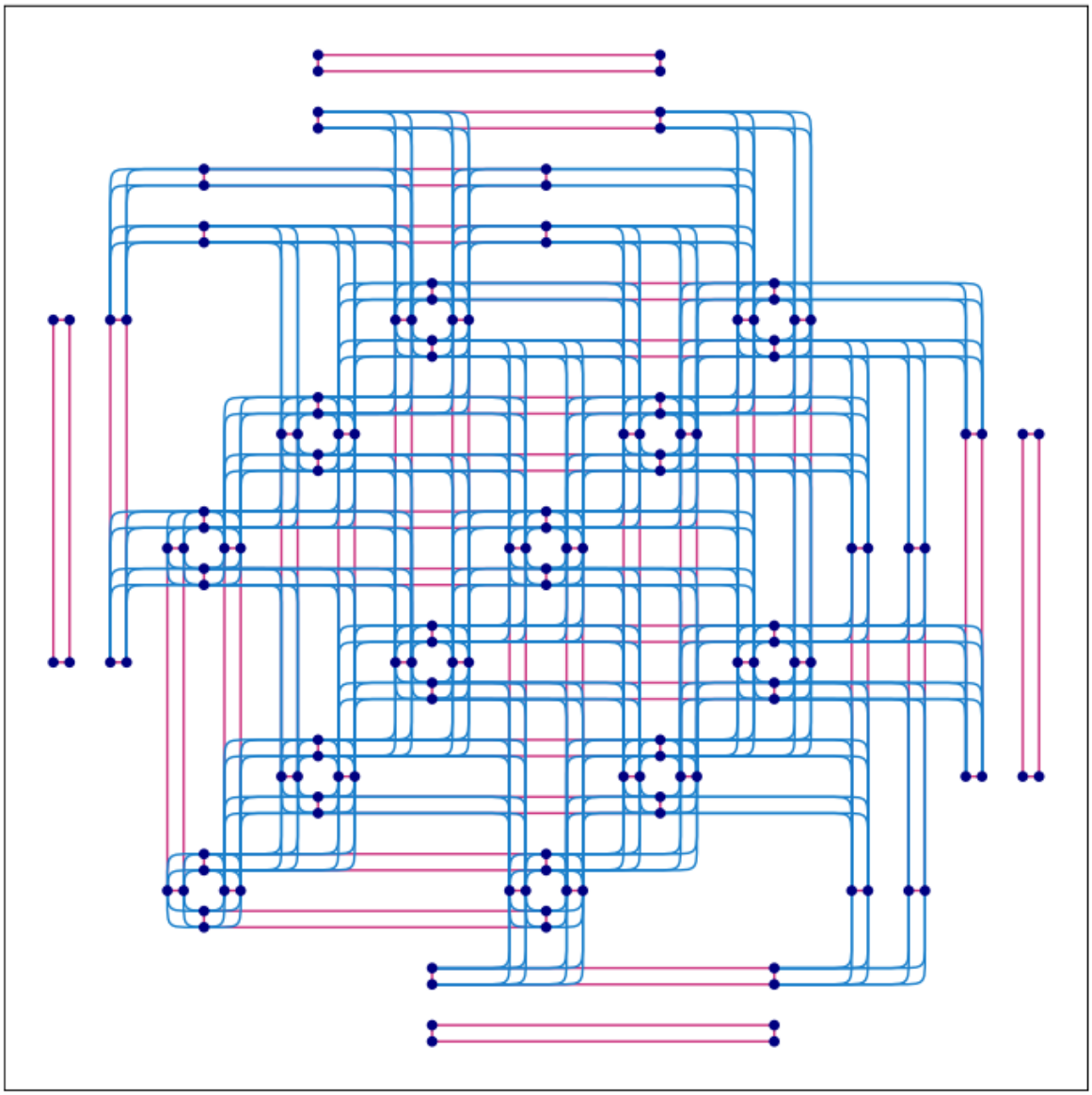
2.5 Η τοπολογία Pegasus

Η τοπολογία «Πήγασος» (“Pegasus”) επιτυγχάνει την μεγαλύτερη διασύνδεση που υπάρχει μέχρι στιγμής στους κβαντικούς υπολογιστές. Με την τοπολογία του Πήγασου κάθε qubit συνδέεται με 15 άλλα qubits, ενώ στην τοπολογία Chimera κάθε qubit συνδέεται με 6 άλλα qubits. Με δυόμισι φορές μεγαλύτερη συνδεσιμότητα, η νέα αυτή τοπολογία επιτρέπει την ενσωμάτωση μεγαλύτερων προβλημάτων με λιγότερα φυσικά qubits από την τοπολογία Chimera.

Όπως στην τοπολογία Chimera χρησιμοποιείται ο συμβολισμός C_n για ένα γράφημα με μέγεθος n, έτσι και στην τοπολογία Pegasus χρησιμοποιείται ο συμβολισμός P_n. Για παράδειγμα, το P_3 είναι ένα γράφημα με 144 κόμβους.

Και στις δύο τοπολογίες, τα qubits είναι διατεταγμένα καθέτως ή οριζοντίως. Στη Chimera υπάρχουν δύο τύποι ζευκτών: «εσωτερικοί» ζεύκτες που συνδέουν ζεύγη ορθογωνίων (με

αντίθετη διάταξη) qubits και «εξωτερικοί» ζεύκτες που συνδέουν ζεύγη qubits που είναι παράλληλα στην ίδια γραμμή ή στήλη. Οι τοπολογίες Pegasus επιπλέον των «εσωτερικών» και «εξωτερικών» ζευκτών, προσθέτουν ένα ακόμη είδος: τους «μονούς» (odd) ζεύκτες. Οι ζεύκτες αυτοί συνδέουν παράλληλα ζεύγη qubits σε γειτονικές σειρές ή στήλες, όπως βλέπουμε και στο Σχήμα 2-4.



Σχήμα 2-4 Επεξεργαστής Pegasus(0)

Οι μπλε γραμμές συμβολίζουν τους «εσωτερικούς» ζεύκτες, οι μακρές κόκκινες τους «εξωτερικούς» ζεύκτες και οι βραχείες κόκκινες γραμμές τους «μονούς» ζεύκτες.

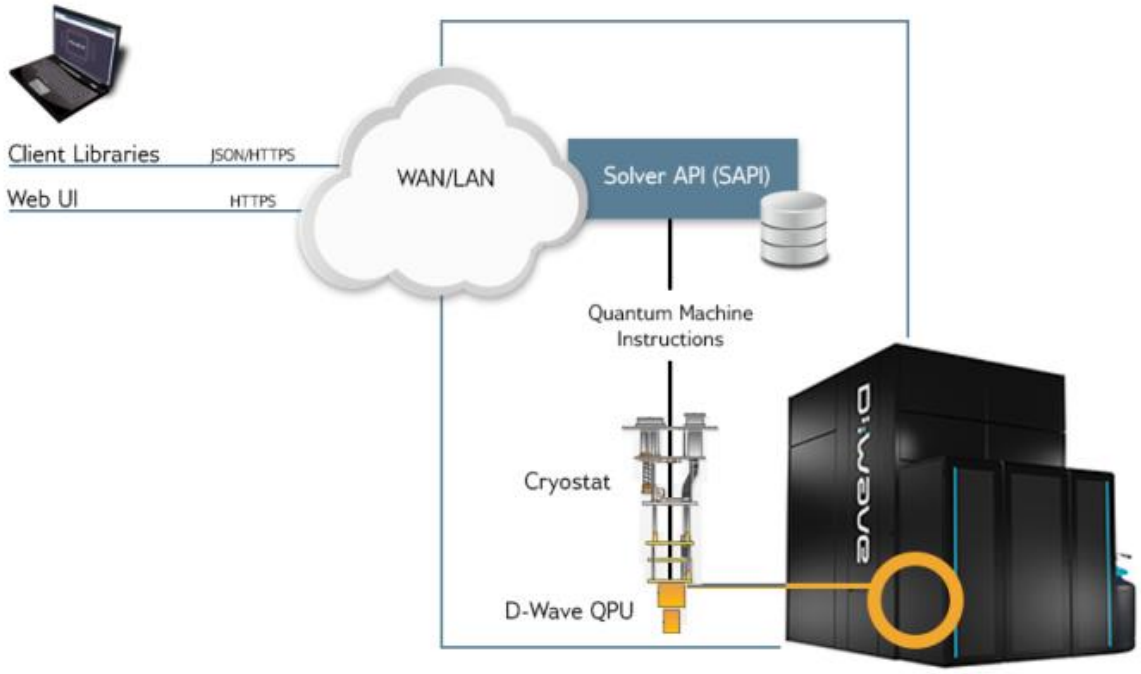
Στην τοπολογία Chimera, θεωρείται ότι τα qubits έχουν ονομαστικό μήκος 4 (κάθε qubit συνδέεται με 4 ορθογώνια qubits μέσω των «εσωτερικών» ζευκτών) και βαθμό 6 (κάθε

qubit συνδέεται με 6 διαφορετικά qubits μέσω ζευκτών). Στην τοπολογία Pegasus, τα qubits έχουν ονομαστικό μήκος 12 και βαθμό 15 (D-Wave Systems Inc., 2019).

2.6 Περιβάλλον προγραμματισμού D-Wave

Οι χρήστες του κβαντικού υπολογιστή D-Wave αλληλοεπιδρούν με αυτόν μέσω διαδικτυακής διεπαφής (web interface), καθώς και μέσω βιβλιοθηκών πελάτη οι οποίες επικοινωνούν με τη διεπαφή προγραμματισμού εφαρμογών Solver API (SAPI).

Η SAPI περιλαμβάνει στοιχεία που είναι υπεύθυνα για την αλληλεπίδραση του χρήστη, την ταυτοποίηση χρηστών και τον προγραμματισμό των εργασιών. Επιπλέον, η SAPI αναλαμβάνει την επικοινωνία με διακομιστές back-end για την αποστολή των προς επίλυση προβλημάτων και την επιστροφή των αποτελεσμάτων από τη QPU (Σχήμα 2-5).



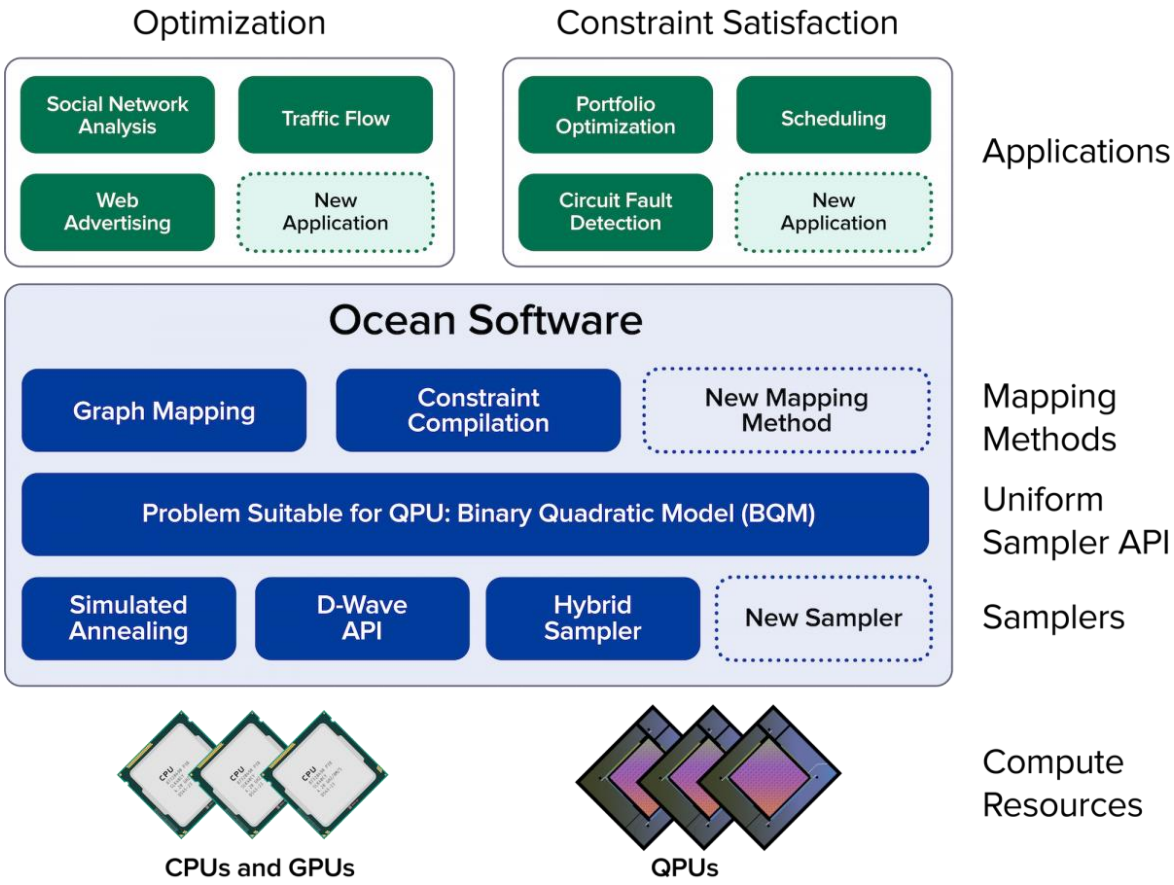
Σχήμα 2-5 Περιβάλλον προγραμματισμού D-Wave

Η χρήση των SAPI web services πραγματοποιείται αξιοποιώντας τα πρωτοκόλλα REST και HTTPS. Τα web services παρέχουν μια σειρά από λειτουργίες, όπως:

- Αναζήτηση διαθέσιμων solvers και των ιδιοτήτων τους
- Αποστολή προβλημάτων προς επίλυση
- Ακύρωση προβλημάτων που είχαν αποσταλεί για επίλυση

- Ανάκτηση κατάστασης προβλήματος
- Λήψη αποτελεσμάτων σε περίπτωση επιτυχούς εκτέλεσης
- Ανάκτηση λαθών σε περίπτωση ανεπιτυχούς εκτέλεσης

Επίσης, η D-Wave παρέχει λογισμικά ανοικτού κώδικα με την ονομασία «Ocean» που συνεισφέρουν στην γρηγορότερη και πιο αποτελεσματική ανάπτυξη εφαρμογών για κβαντικούς υπολογιστές. Πρόκειται δηλαδή για ένα κιτ ανάπτυξης λογισμικού (Software Development Kit – SDK) που είναι δωρεάν διαθέσιμο από τη σελίδα της εταιρίας στο GitHub. Το Ocean SDK περιλαμβάνει λογισμικά και εργαλεία σε γλώσσα Python για τον μετασχηματισμό ενός προβλήματος σε κατάλληλη μορφή που να μπορεί να επιλυθεί από κβαντικό υπολογιστή.



Σχήμα 2-6 Αρχιτεκτονική λογισμικού D-Wave

Όπως βλέπουμε και στο Σχήμα 2-6, τα λογισμικά αυτά αποτελούν το ενδιάμεσο επίπεδο ανάμεσα στις εφαρμογές και τους υπολογιστικούς πόρους:

- Εφαρμογές (Applications): Το αρχικό πρόβλημα που θέλουμε να επιλυθεί, συμπεριλαμβανομένων των απαραίτητων δεδομένων και ενός σαφώς καθορισμένου στόχου.
- Μέθοδοι αντιστοίχισης (Mapping methods): Εργαλεία τα οποία «μεταφράζουν» το στόχο και τα δεδομένα της εφαρμογής σε μορφή προβλήματος κατάλληλη για επίλυση σε κβαντικό υπολογιστή.
- Δειγματολήπτες (Samplers): Εργαλεία τα οποία λαμβάνουν το πρόβλημα και επιστρέφουν πιθανές λύσεις. Το Ocean υλοποιεί αρκετούς δειγματολήπτες που χρησιμοποιούν τη D-Wave QPU, καθώς και κλασικούς υπολογιστικούς πόρους. Στον παρακάτω πίνακα (Πίνακας 2-2) παρουσιάζονται οι διαθέσιμοι δειγματολήπτες.
- Υπολογιστικοί πόροι (Compute resources): Το υλικό στο οποίο γίνεται η επεξεργασία και η λύση του προβλήματος. Μπορεί να είναι η D-Wave QPU, αλλά ακόμη και η CPU του υπολογιστή που γίνεται η ανάπτυξη της εφαρμογής.

Υπολογισμός	**Δειγματολήπτης**	**Χρήση**	**Παρατηρήσεις**
Κλασικός	dimod ExactSampler()	Βρίσκει όλες τις καταστάσεις για μικρά (<20 μεταβλητές) προβλήματα.	Κατάλληλο για δοκιμαστικούς λόγους στη φάση της ανάπτυξης
Κλασικός	dimod RandomSampler()	Τυχαίος δειγματολήπτης για δοκιμές.	>>
Κλασικός	dimod SimulatedAnnealingSampler()	Προσομοίωση δειγματολήπτη ανόπτησης για δοκιμές	>>
Κλασικός	dwave-neal SimulatedAnnealingSampler()	Προσομοίωση δειγματολήπτη ανόπτησης	
Κβαντικός	dwave-system DWaveSampler()	Γρήγορη ενσωμάτωση του συστήματος D-Wave ως δειγματολήπτη	
Κβαντικός	dwave-cloud-client Solver()	Συστήμα D-Wave ως δειγματολήπτης	Για έλεγχο της υποβολής προβλήματος σε χαμηλό επίπεδο.

Κβαντικός	dwave-hybrid KerberosSampler()	Υβριδικός δειγματολήπτης συμβατός με dimod	Για προβλήματα με ανομοιογενή δομή και μέγεθος
	dimod custom	Δυνατότητα δημιουργίας προσαρμοσμένου δειγματολήπτη.	

Πίνακας 2-2 Δειγματολήπτες Ocean

(Πηγή: https://docs.ocean.dwavesys.com/en/latest/overview/stack.html)

Αν και οι χρήστες μπορούν να υποβάλουν προβλήματα προς επίλυση στο σύστημα με διάφορους τρόπους, τελικά ένα πρόβλημα αναπαριστά ένα σύνολο τιμών που αντιστοιχούν στο βάρος των qubits και στην ισχύ των ζευκτών. Η βέλτιστη διαμόρφωση των qubits που βρέθηκε, δηλαδή η κατάσταση της χαμηλότερης ενέργειας, αποτελεί τη λύση του προβλήματος και επιστρέφεται στο χρήστη. Επειδή οι κβαντικοί υπολογιστές είναι πιθανοτικοί και όχι ντετερμινιστικοί, μπορούν να επιστραφούν πολλαπλές τιμές, παρέχοντας όχι μόνο την καλύτερη λύση, αλλά και άλλες πολύ καλές εναλλακτικές λύσεις. Οι χρήστες μπορούν να καθορίσουν τον αριθμό των λύσεων που επιθυμούν να επιστρέψει το σύστημα και γενικότερα η D-Wave δίνει στους χρήστες σημαντικό έλεγχο του κβαντικού υπολογισμού.

Τέλος, η εταιρία D-Wave έχει αναπτύξει τη διαδικτυακή πύλη «Leap» η οποία παρέχει άμεση και δωρεάν δοκιμαστική πρόσβαση σε ένα σύστημα D-Wave 2000Q. Επιπλέον παρέχει χρήσιμο εκπαιδευτικό υλικό, καθώς και τη δυνατότητα εκτέλεσης διαδραστικών παραδειγμάτων με σκοπό την εξοικείωση με τη διαδικασία επίλυσης προβλημάτων στο κβαντικό υπολογιστή D-Wave (D-Wave Systems Inc.).

3. Επίλυση προβλημάτων στον D-Wave

Η ανάγκη για αναπαράσταση συσχετιζόμενων εννοιών οδήγησε στην ανάπτυξη της μαθηματικής οντότητας που ονομάζεται γράφος ή γράφημα (graph). Με τη βοήθεια των γράφων επιτυγχάνεται η απεικόνιση σύνθετων φυσικών καταστάσεων που είναι εξαρτημένοι από πλήθος εννοιών και απαιτούν ένα σημαντικό αριθμό λογικών διαδικασιών. Αν απεικονιστούν οι έννοιες σε σημεία του χώρου και συνδεθούν ανά δύο εκείνα τα σημεία που απεικονίζουν έννοιες που συσχετίζονται, προκύπτει ένας γράφος. Με την ανάπτυξη της επιστήμης των υπολογιστών και της μηχανικής του διαδικτύου, οι γράφοι αποδείχτηκαν ιδιαίτερα σημαντικά βοηθήματα για την επίλυση σύνθετων προβλημάτων (Γεωργίου, Αντωνίου, & Χατζημιχαηλίδης, 2015, p. 137). Η θεωρία γράφων εφαρμόζεται στις τηλεπικοινωνίες, στο σχεδιασμό κυκλωμάτων, στις μεταφορές, στην οικονομία, στο σχεδιασμό και βελτιστοποίηση δικτύων, στη γλωσσολογία, κ.α.

Στο κεφάλαιο αυτό θα ασχοληθούμε με μερικά από τα πιο δημοφιλή NP-πλήρη προβλήματα από τη θεωρία γράφων και πως αυτά μπορούν να προσεγγιστούν χρησιμοποιώντας τον κβαντικό υπολογιστή D-Wave. Ένα πρόβλημα είναι NP-πλήρες, αν δεν υπάρχει αποδοτικός αλγόριθμος που να το επιλύει σε πολυωνυμικό χρόνο (αν $P \neq NP$).

3.1 Εγκατάσταση Ocean SDK

Η εγκατάσταση των εργαλείων «Ocean» μπορεί να πραγματοποιηθεί στα λειτουργικά συστήματα Microsoft Windows (64-bit), macOS και Linux. Απαραίτητη προϋπόθεση είναι να υπάρχει εγκατεστημένο το περιβάλλον προγραμματισμού της Python και συγκεκριμένα έκδοση 3.5 ή νεότερη. Στη συνέχεια ακολουθούν οδηγίες για την εγκατάσταση των εργαλείων σε λειτουργικό σύστημα Microsoft Windows 10.

Αρχικά, εγκαθιστούμε τη γλώσσα προγραμματισμού Python σε έκδοση 64-bit. Για να επαληθεύσουμε ότι η Python έχει εγκατασταθεί επιτυχώς, σε μια γραμμή εντολών (cmd) πληκτρολογούμε:

```
python -V
```

Εντολές 1 Έλεγχος εγκατάστασης Python

Η εντολή θα πρέπει να μας επιστρέψει την έκδοση Python που έχουμε εγκαταστήσει στο σύστημά μας (π.χ. Python 3.6.4). Στη συνέχεια, πρέπει να δημιουργήσουμε ένα

"απομονωμένο" περιβάλλον Python (Virtualenv). Είναι προτιμότερο να εγκαταστήσουμε το Ocean SDK σε ένα τέτοιο περιβάλλον καθώς πρόκειται να εγκαταστήσει αρκετά πακέτα λογισμικών (εξαρτήσεις) και με τον τρόπο αυτό δε θα επηρεαστεί το περιβάλλον προγραμματισμού Python που διαθέτουμε. Με την ακόλουθη εντολή δημιουργούμε το περιβάλλον με όνομα "ocean":

```
python -m venv ocean
```

Εντολές 2 Δημιουργία Virtualenv

Ενεργοποιούμε το περιβάλλον:

```
ocean\Scripts\activate
```

Εντολές 3 Ενεργοποίηση Virtualenv

Καλό θα είναι πριν την εγκατάσταση να προχωρήσουμε σε ενημέρωση του διαχειριστή πακέτων Python pip με την εντολή:

```
python -m pip install --upgrade pip
```

Εντολές 4 Ενημέρωση pip

Πλέον μπορούμε να προχωρήσουμε στην εγκατάσταση του Ocean SDK πληκτρολογώντας την εντολή:

```
pip install dwave-ocean-sdk
```

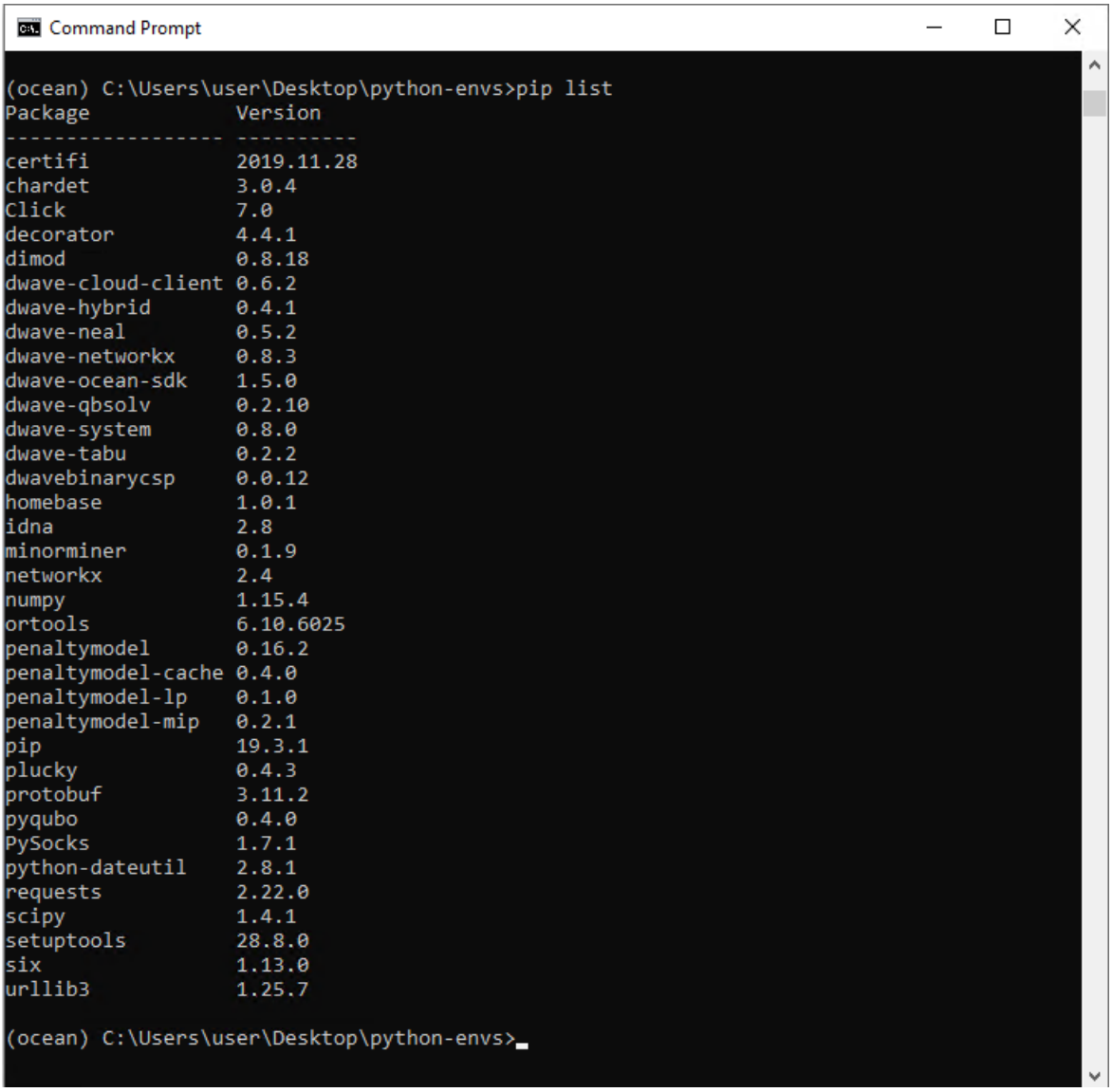
Εντολές 5 Εγκατάσταση Ocean SDK

Επαληθεύουμε ότι η εγκατάσταση πραγματοποιήθηκε με επιτυχία με την εντολή:

```
pip list
```

Εντολές 6 Επαλήθευση εγκατάστασης

Το αποτέλεσμα της παραπάνω εντολής θα πρέπει να περιέχει τη λίστα των πακέτων όπως στην Εικόνα 3-1:



Εικόνα 3-1 Επαλήθευση εγκατάστασης Ocean SDK

Επιπλέον, θα εγκαταστήσουμε τη βιβλιοθήκη Matplotlib που θα τη χρησιμοποιήσουμε για να έχουμε γραφική απεικόνιση των αποτελεσμάτων:

```
pip install matplotlib
```

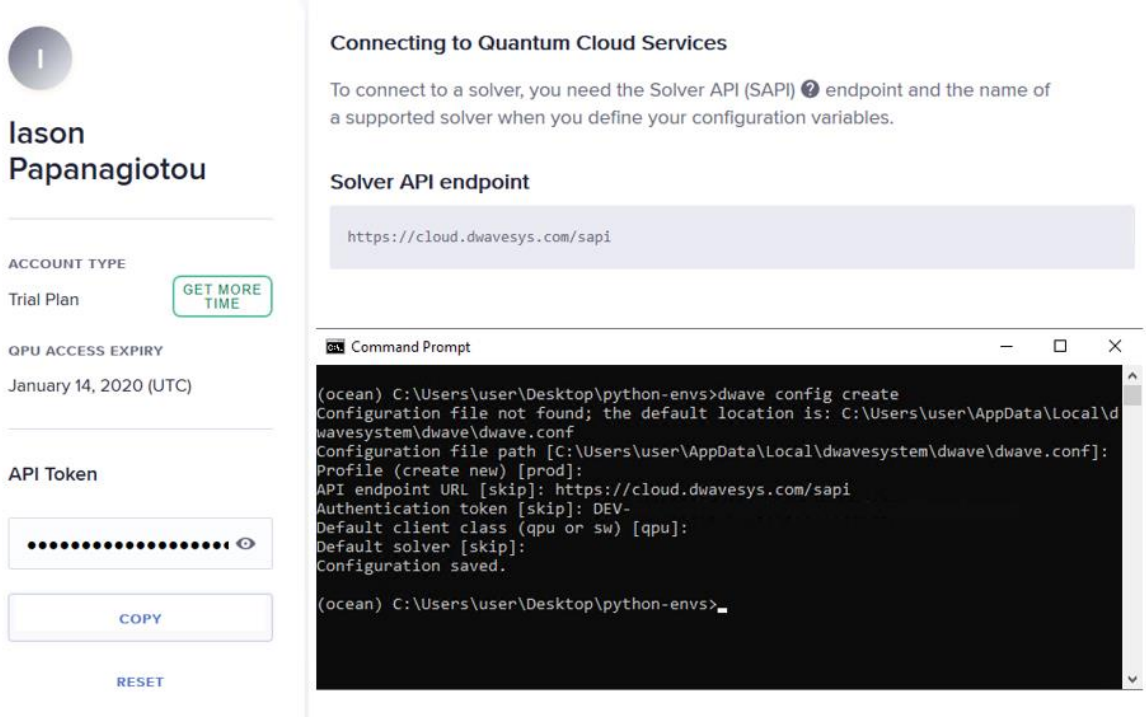
Εντολές 7 Εγκατάσταση βιβλιοθήκης Matplotlib

Τέλος, θα πρέπει να γίνουν κάποιες ρυθμίσεις για την επικοινωνία με το κβαντικό υπολογιστή D-Wave. Πληκτρολογούμε την παρακάτω εντολή η οποία θα μας καθοδηγήσει στη δημιουργία του αρχείου διαμόρφωσης (configuration):

```
dwave config create
```

Εντολές 8 Δημιουργία configuration

Επιλέγουμε τις προεπιλεγμένες τιμές πατώντας το πλήκτρο «Enter» εκτός από τις ρυθμίσεις που αφορούν τα «API endpoint URL» και «Authentication token». Τις πληροφορίες αυτές μπορούμε να τις αντιγράψουμε από το περιβάλλον Leap με τη χρήση του λογαριασμού που έχουμε δημιουργήσει στη D-Wave, όπως βλέπουμε και στην παρακάτω Εικόνα 3-2:



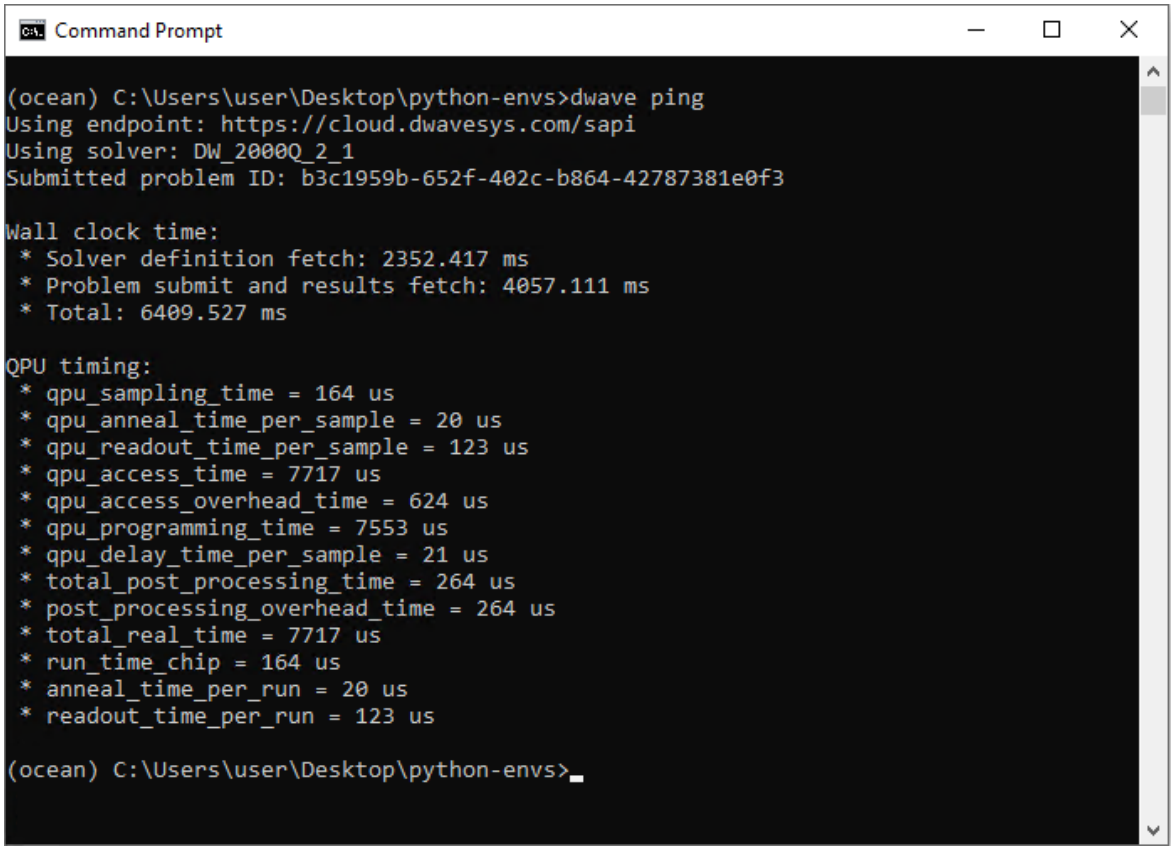
Εικόνα 3-2 LEAP: API URL, Authenitcation Token

Μπορούμε να επαληθεύσουμε την ορθότητα των ρυθμίσεων πληκτρολογώντας την παρακάτω εντολή η οποία στέλνει ένα απλό πρόβλημα προς επίλυση:

```
dwave ping
```

Εντολές 9 Επαλήθευση επικοινωνίας με D-Wave

Το αποτέλεσμα θα πρέπει να είναι παρόμοιο με αυτό στην Εικόνα 3-3



Εικόνα 3-3 Επιτυχής αποστολή προβλήματος στον D-Wave

3.2 Το πρόβλημα του μέγιστου συνόλου ανεξαρτησίας

Ένα σύνολο ανεξαρτησίας (independent set) ενός μη κατευθυντικού γραφήματος είναι ένα υπογράφημα / υποσύνολο κορυφών στο οποίο κανένα ζευγάρι κορυφών δεν συνδέεται με ακμή. Δεδομένου ενός μη κατευθυντικού γραφήματος *G(V, E)* και ενός ακέραιου $K \geq 1$, η εκδοχή απόφασης του προβλήματος του Μέγιστου Συνόλου Ανεξαρτησίας (MIS – Maximum Independent Set) εξετάζει αν το *G* περιέχει ένα σύνολο ανεξαρτησίας μεγέθους τουλάχιστον *K* (Φωτάκης & Σπυράκης, 2001, σ. 186).

Δηλαδή, ένα σύνολο ανεξαρτησίας είναι μέγιστο (maximum), όταν δεν υπάρχει άλλο μεγαλύτερο σύνολο ανεξαρτησίας στο γράφημα. Ο αριθμός των κορυφών του μεγαλύτερου συνόλου ανεξαρτησίας ονομάζεται αριθμός ανεξαρτησίας (independence number) και συμβολίζεται με $\alpha(G)$. Ένα γράφημα μπορεί να έχει πολλά μέγιστα σύνολα ανεξαρτησίας.

Ακολουθεί ο κώδικας σε γλώσσα Python για την επίλυση του προβλήματος:

```
## Εισαγωγή αρθρώματος sys
```

```
import sys

## Εισαγωγή αρθρώματος networkx για το χειρισμό γραφημάτων
import networkx as nx

## Εισαγωγή αρθρώματος dwave_networkx της D-Wave
import dwave_networkx as dnx

## Εισαγωγή αρθρώματος matplotlib.pyplot για την απεικόνιση των γραφημάτων
import matplotlib.pyplot as plt

## Εισαγωγή αρθρώματος - ορισμός πόρου για την επίλυση του προβλήματος
from dwave.system.samplers import DWaveSampler
from dwave.system.composites import EmbeddingComposite
sampler = EmbeddingComposite(DWaveSampler())

## Δημιουργία ενός κενού γραφήματος
G = nx.Graph()

## Προσθήκη κορυφών και ακμών στο γράφημα
#G1
G.add_nodes_from([1,2,3,4,5])
G.add_edges_from([(1,5),(2,5),(2,3),(5,3),(4,3)])

#G2
#G.add_nodes_from([1,2,3,4,5,6,7])
#G.add_edges_from([(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),
#                  (2,3),(3,4),(4,5),(5,6),(6,7),(7,2)])

## Απεικόνιση του γραφήματος πριν την αποστολή για επίλυση
plt.figure()
nx.draw(G,pos=nx.spring_layout(G),with_labels=True)
plt.show()

## Έλεγχος για συνέχεια του προγράμματος ανάλογα την απάντηση του χρήστη
yes=["yes","y","YES","Y"]
no=["no","n","N","NO"]
answer = ""
```

```
while answer not in yes:
    print("Θέλετε να συνεχίσετε; [Y/n]")
    answer = input()
    if answer in no:
        sys.exit("Το πρόγραμμα τερματίστηκε")



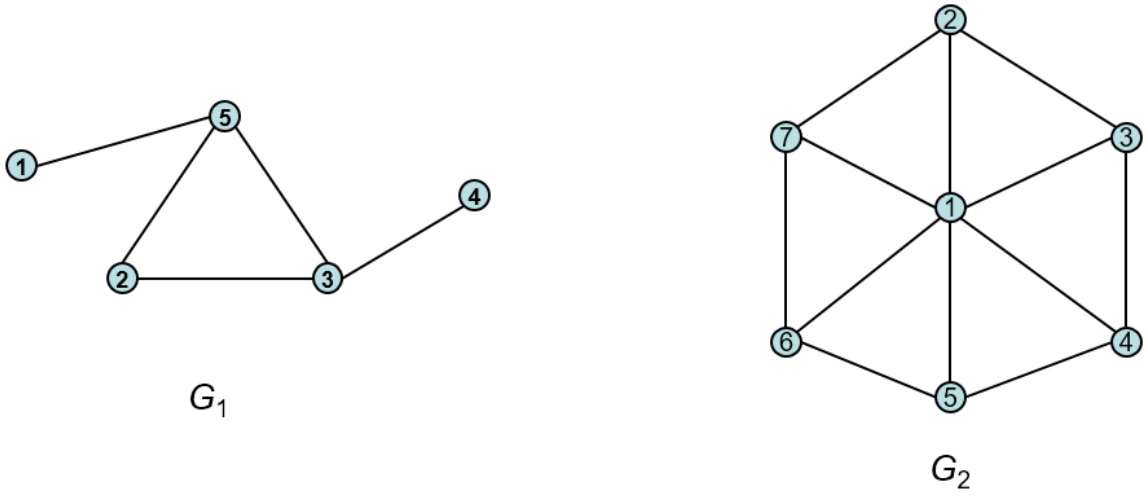
## Εύρεση μέγιστου συνόλου ανεξαρτησίας
S = dnx.maximum_independent_set(G, sampler=sampler, num_reads=10)

## Εκτύπωση της λύσης
print('Ο αριθμός ανεξαρτησίας του γραφήματος είναι: ', len(S))
print('Το μέγιστο σύνολο ανεξαρτησίας αποτελείται από τις κορυφές: ', S)

## Απεικόνιση της λύσης
## Οι κορυφές με πράσινο χρώμα ανήκουν στο μέγιστο σύνολο ανεξαρτησίας
k = G.subgraph(S)
notS = list(set(G.nodes())-set(S))
othersubgraph = G.subgraph(notS)
pos = nx.spring_layout(G)
plt.figure()
nx.draw(G,pos=pos,with_labels=True)
nx.draw(k,pos=pos,with_labels=True, node_color='green')
nx.draw(othersubgraph,pos=pos,with_labels=True)
plt.show()
```

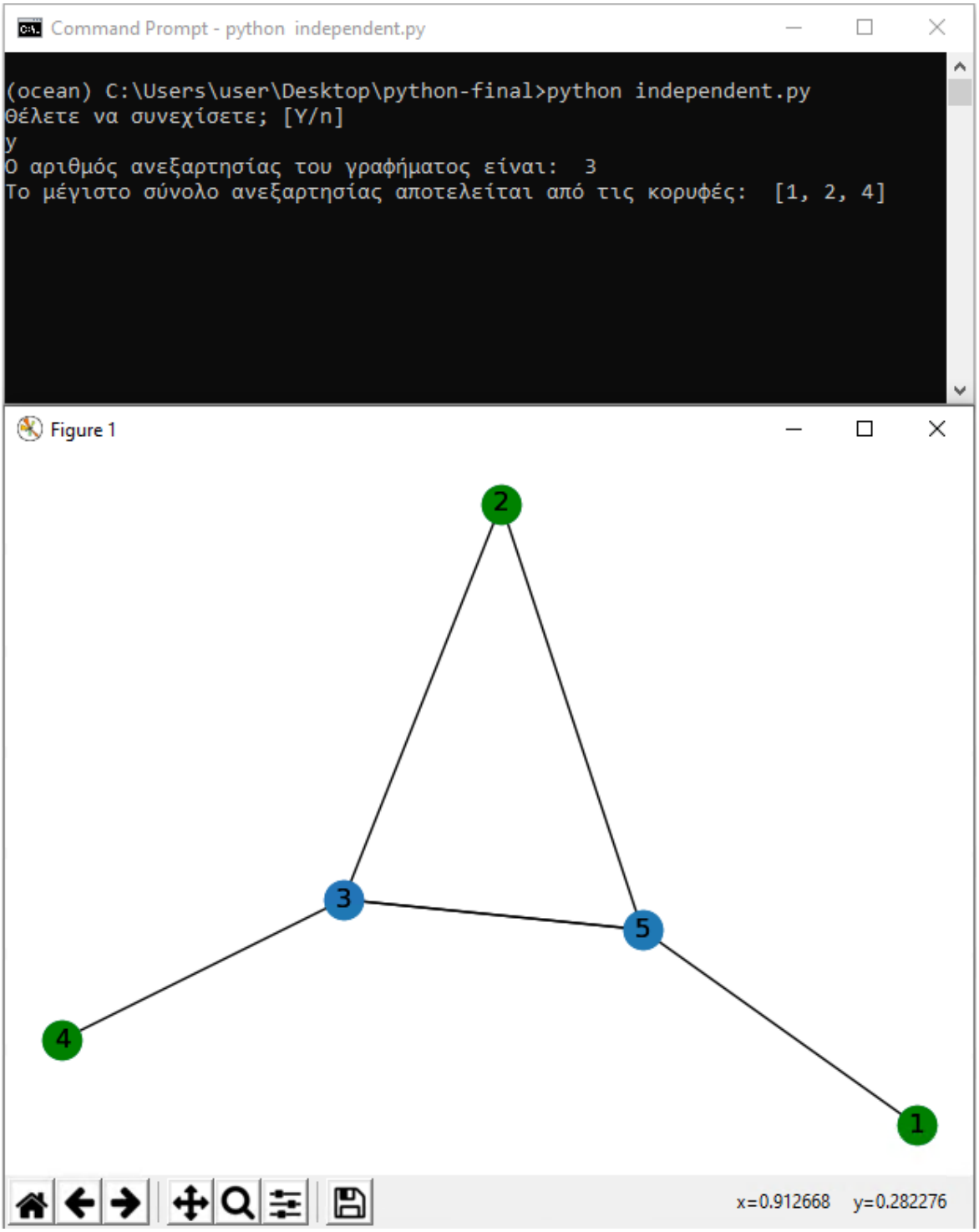
Κώδικας 1 Μέγιστο σύνολο ανεξαρτησίας

Στη συνέχεια ακολουθεί παράδειγμα εκτέλεσης του προγράμματος για τους γράφους G_1, G_2 (Σχήμα 3-1):

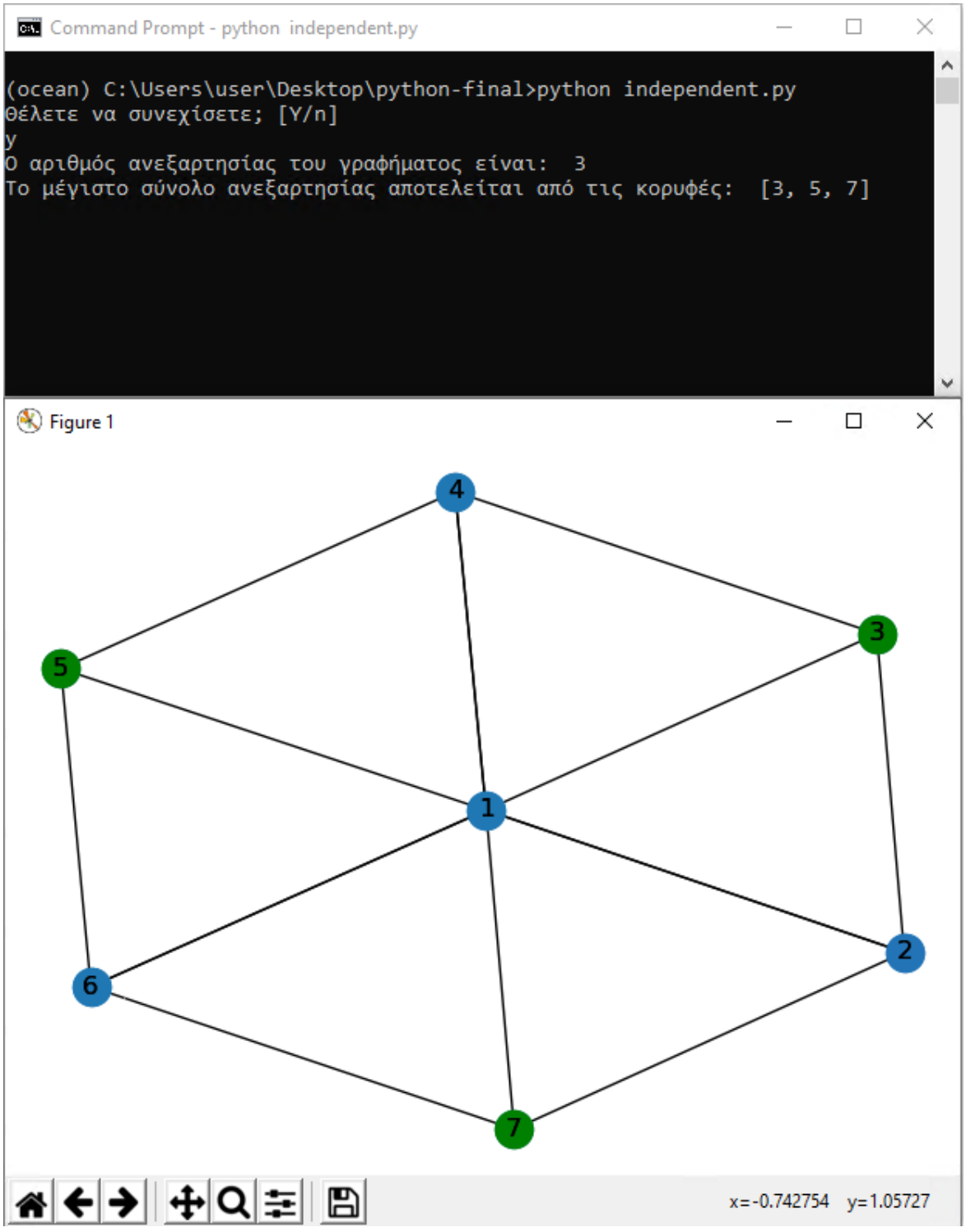


Σχήμα 3-1 Γράφοι G_1, G_2

Τα αποτελέσματα εκτέλεσης του προγράμματος παρουσιάζονται στην Εικόνα 3-4 και Εικόνα 3-5, αντίστοιχα. Με πράσινο χρώμα σημειώνονται οι κορυφές που ανήκουν στο μέγιστο σύνολο ανεξαρτησίας.



Εικόνα 3-4 ΜΣΑ για το γράφημα G_1



Εικόνα 3-5 ΜΣΑ για το γράφημα G_2

3.3 Το πρόβλημα του χρωματικού αριθμού γραφήματος

Ένας έγκυρος χρωματισμός ενός γραφήματος *G* ονομάζεται μια ανάθεση χρωμάτων στις κορυφές του, ούτως ώστε κάθε ζευγάρι κορυφών που συνδέεται με ακμή να έχει διαφορετικό χρώμα. Ο χρωματικός αριθμός ενός γραφήματος είναι ο ελάχιστος αριθμός χρωμάτων για τον οποίο υπάρχει ένας έγκυρος χρωματισμός και συμβολίζεται με χ(*G*) (Φωτάκης Δ.).

Ακολουθεί ο κώδικας σε γλώσσα Python για την επίλυση του προβλήματος:

```
## Εισαγωγή αρθρώματος sys
import sys

## Εισαγωγή αρθρώματος networkx για το χειρισμό γραφημάτων
import networkx as nx

## Εισαγωγή αρθρώματος dwave_networkx της D-Wave
import dwave_networkx as dnx

## Εισαγωγή αρθρώματος matplotlib.pyplot για την απεικόνιση των γραφημάτων
import matplotlib.pyplot as plt

## Εισαγωγή αρθρώματος - ορισμός πόρου για την επίλυση του προβλήματος
from hybrid.reference.kerberos import KerberosSampler
sampler = KerberosSampler()

## Δημιουργία ενός κενού γραφήματος
G = nx.Graph()

## Προσθήκη κορυφών (με αύξουσα σειρά) και ακμών στο γράφημα
#G3
#G.add_nodes_from([1,2,3,4,5,6])
#G.add_edges_from([(2,1),(5,1),(6,1),(1,4),(1,3),(2,3),(3,4),(3,5),(3,6),(2,4),(4,5),(5,6)])

#G4
#G.add_nodes_from([1,2,3,4,5,6])
#G.add_edges_from([(1,3),(1,2),(2,3),(2,4),(2,6),(3,4),(3,6),(3,5),(4,6),(5,6)])
```

```
#Greece
G.add_nodes_from([1,2,3,4,5,6,7,8,9,10,11,12,13])
G.add_edges_from([(1,2),(1,11),(2,11),(2,3),(2,5),(3,4),(3,5),(4,5),(4,6),(4,7),(4,8),(5,7),(5,8),(5,11),(6,7),(7,8),(7,10),(8,9),(8,11),(8,12),(9,10),(9,12),(9,13),(11,12),(12,13),(10,13)])


## Απεικόνιση του γραφήματος πριν την αποστολή για επίλυση
plt.figure()
nx.draw(G,pos=nx.spring_layout(G),with_labels=True)
plt.show()

## Έλεγχος για συνέχεια του προγράμματος ανάλογα την απάντηση του χρήστη
yes=["yes","y","YES","Y"]
no=["no","n","N","NO"]
answer = ""
while answer not in yes:
    print("Θέλετε να συνεχίσετε; [Y/n]")
    answer = input()
    if answer in no:
        sys.exit("Το πρόγραμμα τερματίστηκε")



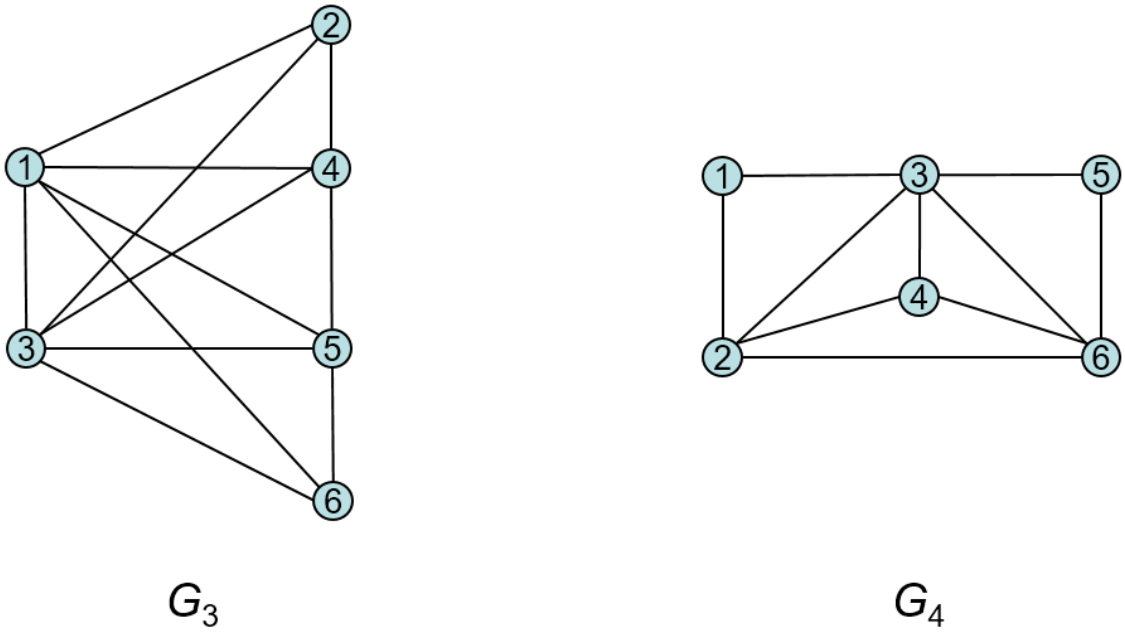
## Εύρεση χρωματισμού κορυφών χρησιμοποιώντας τον ελάχιστο αριθμό χρωμάτων
S = dnx.min_vertex_color(G, sampler=sampler)

## Εκτύπωση της λύσης
## Get dictionary values, find max, add 1
print('Ο χρωματικός αριθμός του γραφήματος είναι: ', max(S.values())+1)
print("Κορυφές:", list(S.keys()))
print("  Χρώμα:", list(S.values()))

## Απεικόνιση της λύσης
plt.figure()
nx.draw(G,pos=nx.spring_layout(G),with_labels=True,node_color=list(S.values()))
plt.show()
```

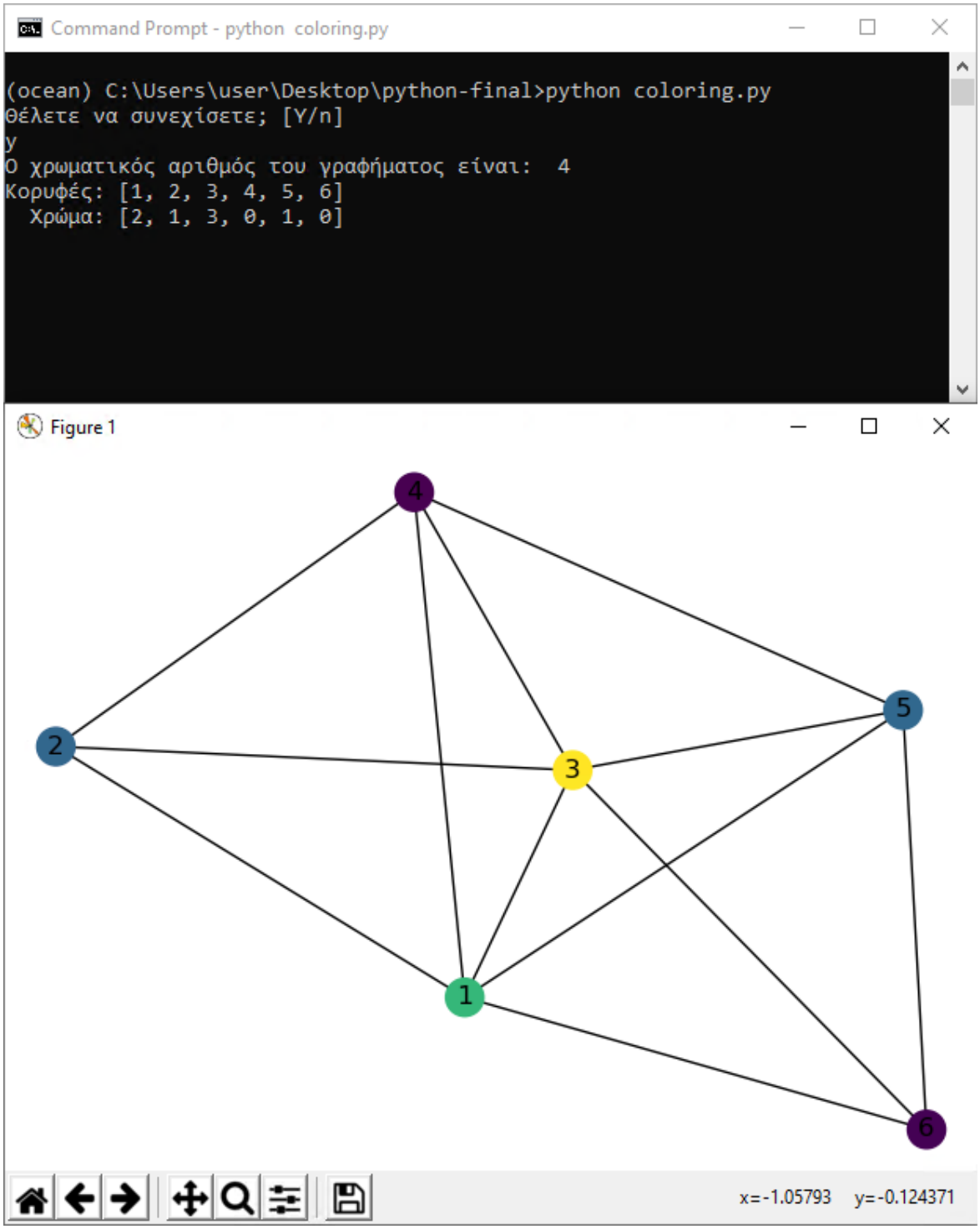
Κώδικας 2 Εύρεση χρωματικού αριθμού

Στη συνέχεια ακολουθεί παράδειγμα εκτέλεσης του προγράμματος για τους γράφους G_3, G_4 (Σχήμα 3-2):

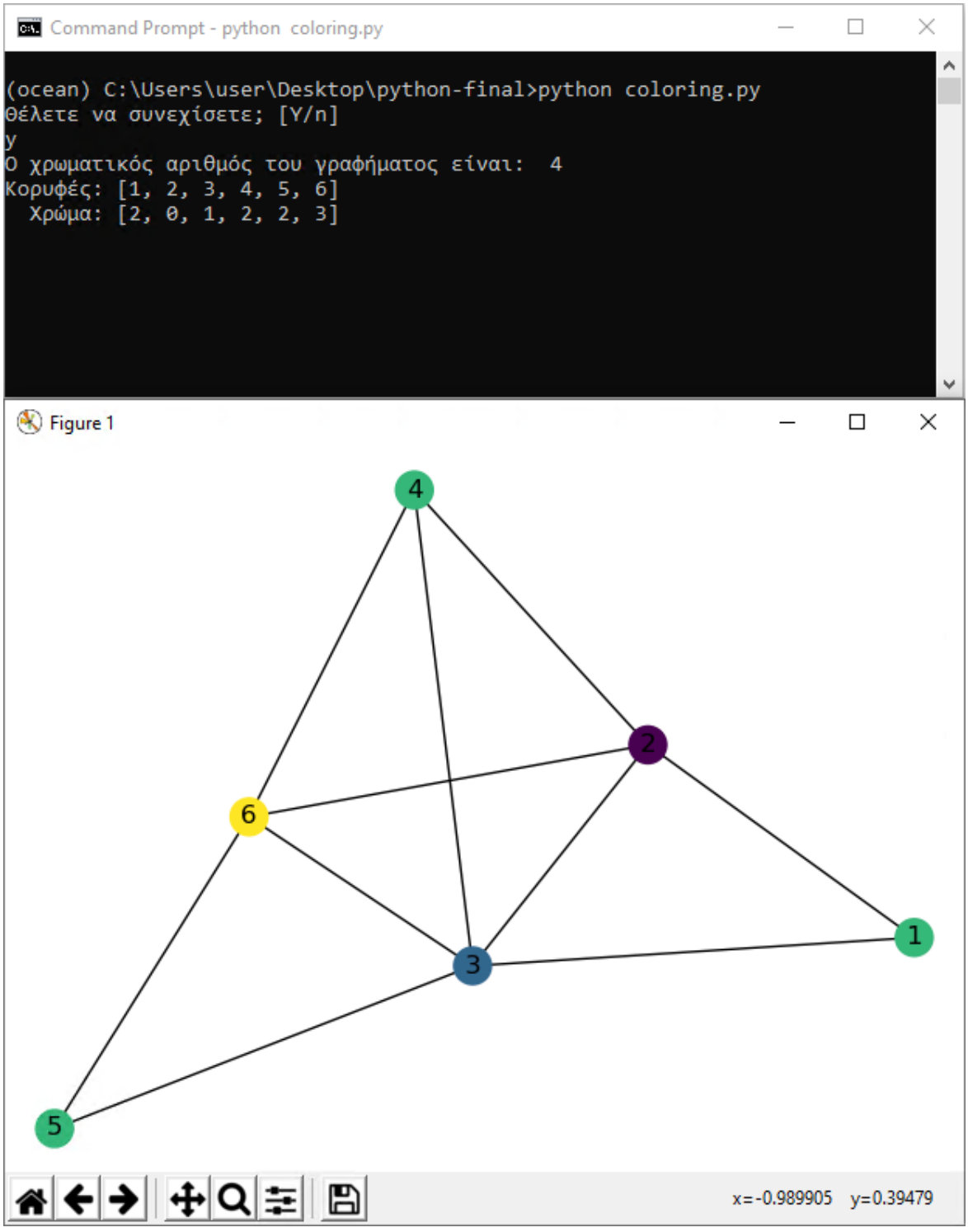


Σχήμα 3-2 Γράφοι G_3, G_4

Τα αποτελέσματα εκτέλεσης του προγράμματος παρουσιάζονται στην Εικόνα 3-6 και Εικόνα 3-7 αντίστοιχα.



Εικόνα 3-6 Χρωματισμός γραφήματος G_3



Εικόνα 3-7 Χρωματισμός γραφήματος G_4

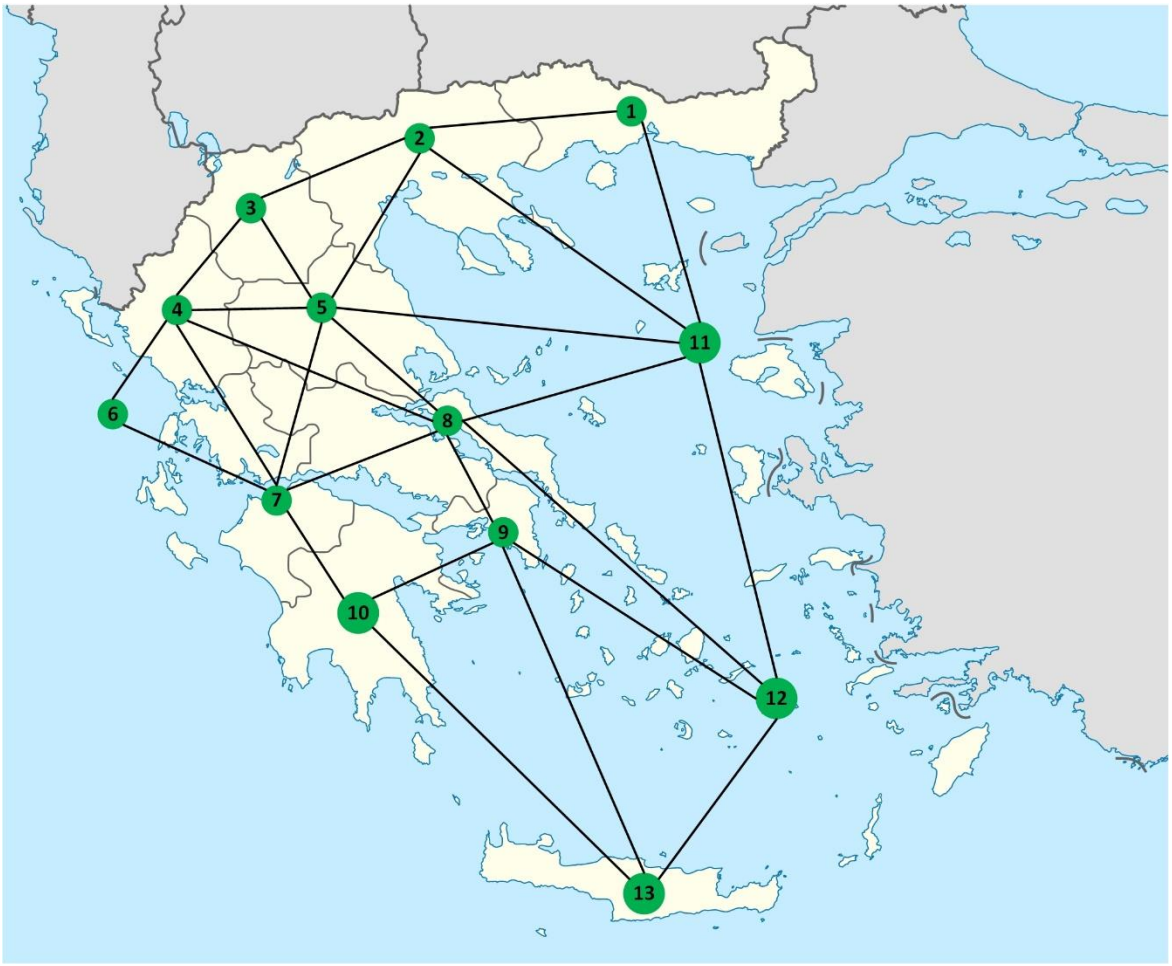
3.4 Το πρόβλημα του χρωματισμού χάρτη

Το πρόβλημα χρωματισμού χάρτη είναι να βρεθεί το ελάχιστο πλήθος χρωμάτων που απαιτούνται για να χρωματισθεί ένας χάρτης, έτσι ώστε δύο γειτονικές περιοχές να μην έχουν το ίδιο χρώμα.

Ο χρωματισμός γραφήματος, -όπως είδαμε και προηγουμένως-, είναι το πρόβλημα της εκχώρησης ενός χρώματος στις κορυφές ενός γράφου με τέτοιο τρόπο, ώστε γειτονικές κορυφές να μην έχουν το ίδιο χρώμα.

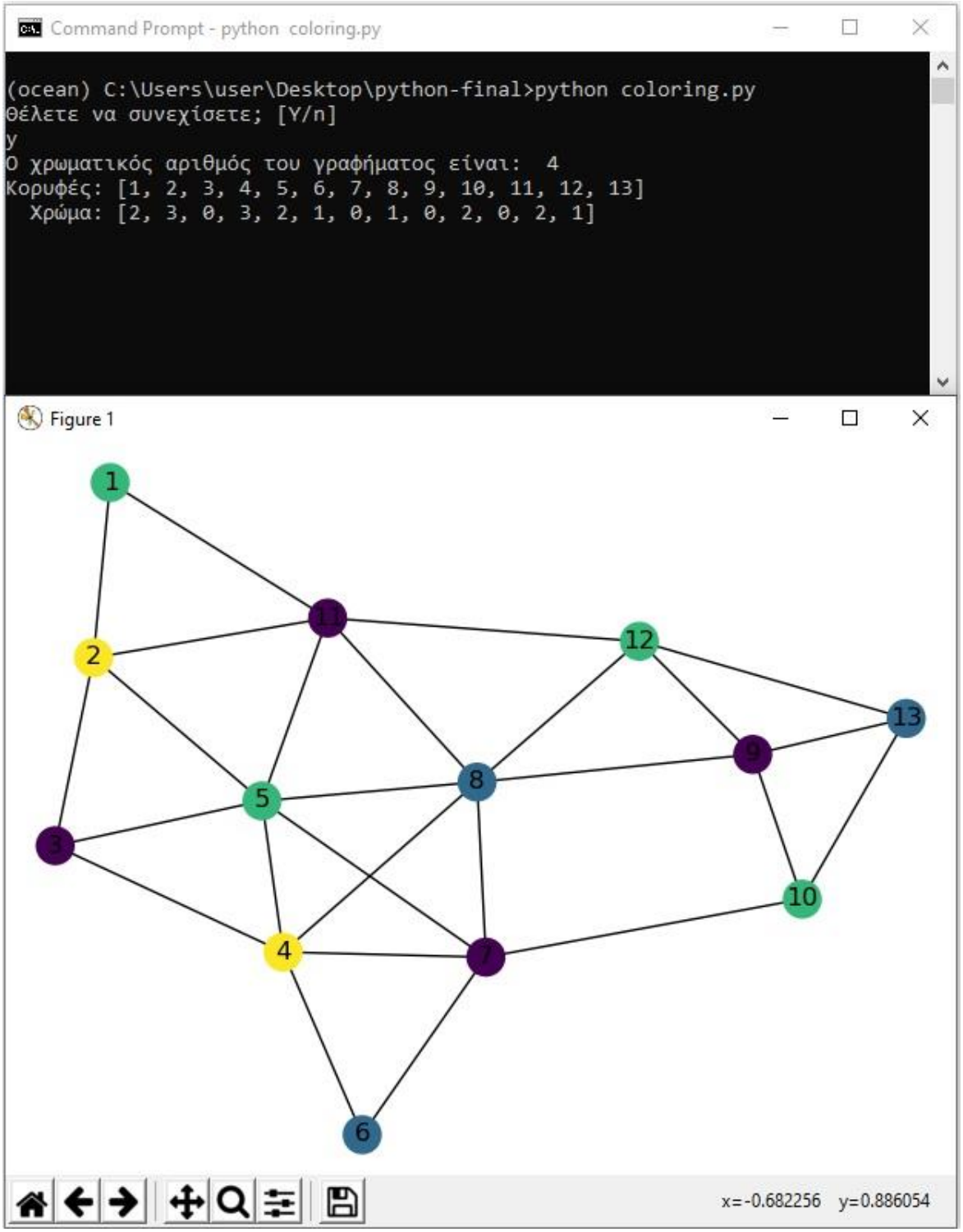
Συνεπώς, το πρόβλημα χρωματισμού χάρτη είναι να αντιστοιχηθεί ένα χρώμα σε κάθε περιοχή ενός χάρτη (οι οποίες μπορούν να αναπαρασταθούν ως μια κορυφή γραφήματος), έτσι ώστε οποιεσδήποτε δύο περιοχές που μοιράζονται ένα όριο (το οποίο μπορεί να αναπαρασταθεί με μια ακμή στο γράφημα) να έχουν διαφορετικά χρώματα.

Στη συνέχεια ακολουθεί παράδειγμα χρωματισμού του ελληνικού χάρτη βάσει της διοικητικής διαίρεσης της Ελλάδας σε περιφέρειες. Αρχικά κατασκευάζουμε το γράφο εισάγοντας μια κορυφή για κάθε περιφέρεια και ακμή για κάθε ζεύγος με κοινά σύνορα (Σχήμα 3-3).

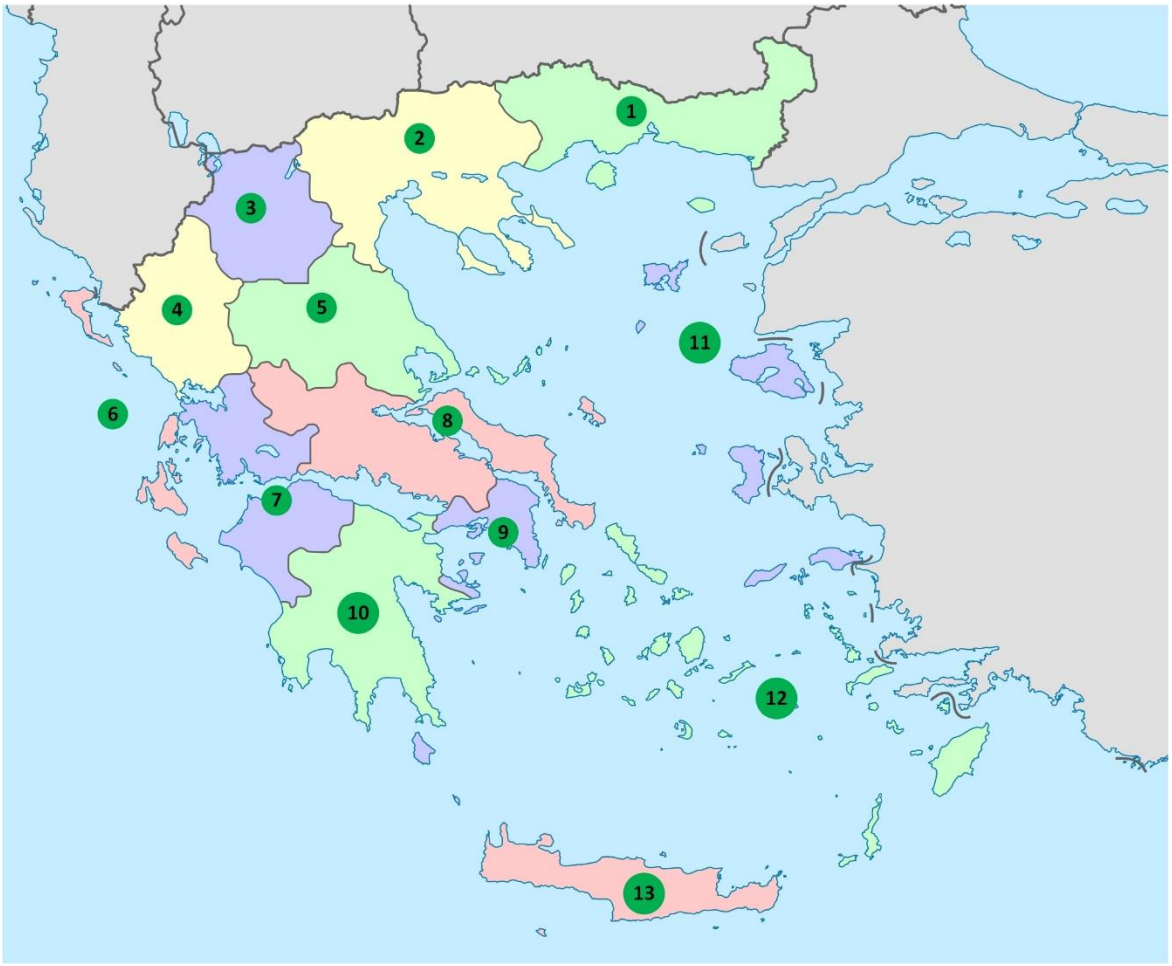


Σχήμα 3-3 Οι διοικητικές περιφέρειες της Ελλάδας ως γράφος

Στη συνέχεια χρησιμοποιούμε το πρόγραμμα της ενότητας 3.3 προσθέτοντας τις κορυφές και τις ακμές του γραφήματος. Το αποτέλεσμα εκτέλεσης του προγράμματος παρουσιάζεται στην Εικόνα 3-8 και Εικόνα 3-9.



Εικόνα 3-8 Χρωματισμός γράφου «Ελλάδα»



Εικόνα 3-9 Χρωματισμός χάρτη Ελλάδας

3.5 Το πρόβλημα του πλανόδιου πωλητή (TSP)

Με δεδομένο ένα σύνολο πόλεων με τις αντίστοιχες χιλιομετρικές τους αποστάσεις, σκοπός είναι να βρεθεί η συντομότερη διαδρομή που επισκέπτεται κάθε πόλη ακριβώς μια φορά και επιστρέφει στην πόλη προέλευσης. Το πρόβλημα του πλανόδιου πωλητή (Travelling Salesman Problem – TSP) συνίσταται στην προσπάθεια του πωλητή να επισκεφτεί όλες τις πόλεις με το ελάχιστο κόστος (άθροισμα αποστάσεων), μια μόνο φορά την κάθε μία, και να επιστρέψει στην πόλη από την οποία ξεκίνησε.

Το πρόβλημα μπορεί να μοντελοποιηθεί με μη κατευθυνόμενο γράφο *G(V, E)* με βάρη στις ακμές, όπου:

- οι πόλεις αποτελούν τις κορυφές
- αν υπάρχει διαδρομή που συνδέει δύο πόλεις, τότε εισάγεται μια ακμή

- κάθε ακμή έχει θετικό βάρος ίσο με την χιλιομετρική απόσταση ανάμεσα στις πόλεις

Το TSP είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης καθώς αναζητάμε την ύπαρξη Χαμιλτονιανών κύκλων στο γράφημα και από αυτούς αυτόν με το μικρότερο κόστος (Lucas, 2014).

Ακολουθεί ο κώδικας σε γλώσσα Python για την επίλυση του προβλήματος:

```
## Εισαγωγή αρθρώματος sys
import sys

## Εισαγωγή αρθρώματος networkx για το χειρισμό γραφημάτων
import networkx as nx

## Εισαγωγή αρθρώματος dwave_networkx της D-Wave
import dwave_networkx as dnx

## Εισαγωγή αρθρώματος matplotlib.pyplot για την απεικόνιση των γραφημάτων
import matplotlib.pyplot as plt

## Εισαγωγή αρθρώματος - ορισμός πόρου για την επίλυση του προβλήματος
from dwave.system.samplers import DWaveSampler
from dwave.system.composites import EmbeddingComposite
sampler = EmbeddingComposite(DWaveSampler())

## Δημιουργία ενός κενού γραφήματος
G = nx.Graph()

## Προσθήκη κορυφών, ακμών και βαρών στο γράφημα
#G5
#G.add_weighted_edges_from([(1,2,10),(1,3,35),(1,4,30),(2,3,30),(2,4,15),(3,4,30)])


#G6
G.add_weighted_edges_from([(1,2,12),(1,3,14),(1,4,17),(2,3,15),(2,4,18),(3,4,29)])


## Απεικόνιση του γραφήματος πριν την αποστολή για επίλυση
plt.figure()
pos = nx.spring_layout(G)
```

```
edge_weights = nx.get_edge_attributes(G, "weight")
#print(edge_weights)
nx.draw(G, pos, with_labels=True)
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_weights)
plt.show()

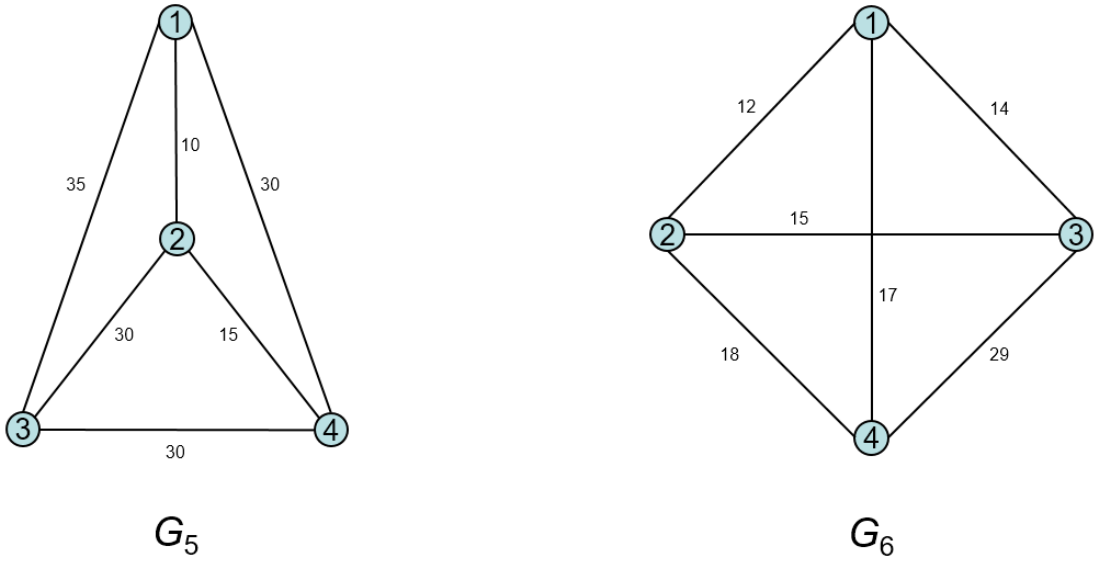
## Έλεγχος για συνέχεια του προγράμματος ανάλογα την απάντηση του χρήστη
yes=["yes","y","YES","Y"]
no=["no","n","N","NO"]
answer = ""
while answer not in yes:
    print("Θέλετε να συνεχίσετε; [Y/n]")
    answer = input()
    if answer in no:
        sys.exit("Το πρόγραμμα τερματίστηκε")


## Εύρεση βέλτιστης διαδρομής
S = dnx.traveling_salesperson(G, sampler=sampler)

## Εκτύπωση της λύσης
print('Η βέλτιστη διαδρομή είναι: ', S)
```

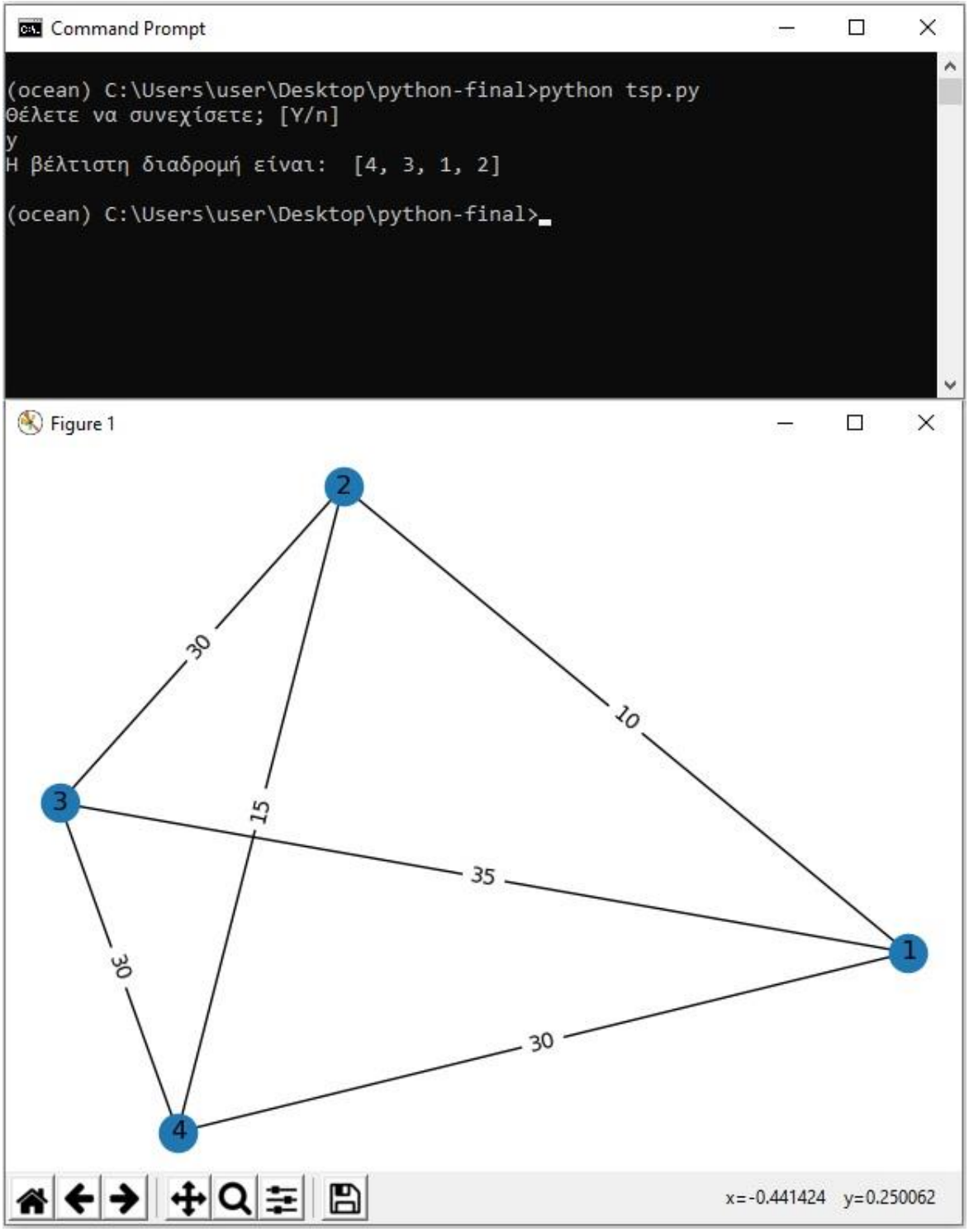
Κώδικας 3 Πρόβλημα πλανόδιου πωλητή

Στη συνέχεια ακολουθεί παράδειγμα εκτέλεσης του προγράμματος για τους γράφους G_5, G_6 (Σχήμα 3-4):

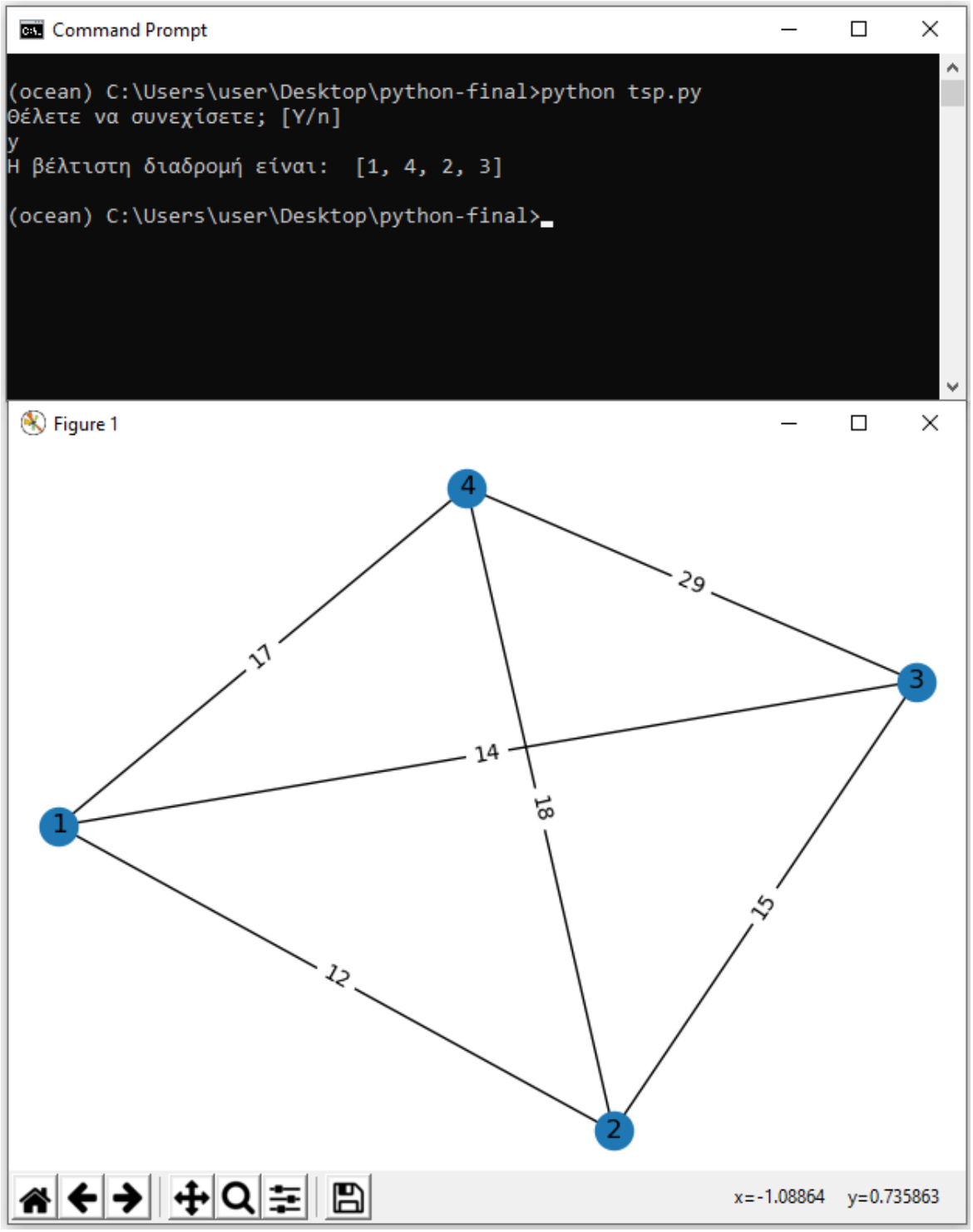


Σχήμα 3-4 Γράφοι G_5, G_6

Τα αποτελέσματα εκτέλεσης του προγράμματος παρουσιάζονται στην Εικόνα 3-10 και Εικόνα 3-11 αντίστοιχα.



Εικόνα 3-10 TSP για το γράφημα G_5



Εικόνα 3-11 TSP για το γράφημα G_6

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

Bennett, C. H., & DiVincenzo, D. P. (2000). Quantum information and computation. Nature, 404, 247-255.

Bertoni, A., Bordone, P., Brunetti, R., Jacoboni, C., & Reggiani, S. (2000). Quantum logic gates based on coherent electron transport in quantum wires. Physical Review Letters, 84(25), 5912-5915.

DiVincenzo, D. P. (2000). The Physical Implementation of Quantum Computation. Fortschritte der Physik: Progress of Physics, 48(9-11), 771-783.

D-Wave Systems Inc. (2018). Getting Started with the D-Wave System. Ανάκτηση Νοέμβριος 25, 2019, από https://docs.dwavesys.com/docs/latest/_downloads/09-1076A-T_GettingStarted.pdf

D-Wave Systems Inc. (2019). Next-Generation Topology of D-Wave Quantum Processors. Ανάκτηση Δεκέμβριος 7, 2019, από https://www.dwavesys.com/sites/default/files/14-1026A-C_Next-Generation-Topology-of-DW-Quantum-Processors.pdf

D-Wave Systems Inc. (n.d.). The D-Wave 2000 Quantum Computer Technology Overview. Ανάκτηση Δεκέμβριος 8, 2019, από https://www.dwavesys.com/sites/default/files/D-Wave%202000Q%20Tech%20Collateral_1029F.pdf

Kalai, G. (2016). The Quantum Computer Puzzle. Notices of the AMS, 63(5), 508-516.

Lucas, A. (2014). Ising formulations of many NP problems. Frontiers in Physics 2:5. doi:10.3389/fphy.2014.00005

Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information. New York: Cambridge University Press.

Pakin, S., Dahl, D., & Ewald, B. (2017). An Introduction to Quantum Computing, D-Wave Style. Ανάκτηση Δεκέμβριος 7, 2019, από https://www.dwavesys.com/sites/default/files/Introduction%20to%20Quantum%20Computing%20-%20Part%201.pdf

Βικιπαίδεια. (2019). Κβαντικός υπολογιστής. Ανάκτηση Νοέμβριος 2019, από Βικιπαίδεια: https://el.wikipedia.org/wiki/%CE%9A%CE%B2%CE%B1%CE%BD%CF%84%CE%B9

%CE%BA%CF%8C%CF%82_%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%AE%CF%82

Γεωργίου, Δ., Αντωνίου, Ε., & Χατζημιχαηλίδης, Α. (2015). Διακριτές μαθηματικές δομές για την επιστήμη των υπολογιστών. Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Ανάκτηση Ιανουάριος 20, 2020, από http://hdl.handle.net/11419/457

Καραφυλλίδης, Ι. (2015). Κβαντική υπολογιστική. Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών (ΣΕΑΒ). Ανάκτηση Νοέμβριος 28, 2019, από https://repository.kallipos.gr/handle/11419/216

Φωτάκης, Δ. (n.d.). Θεωρία Γραφημάτων: Σύνολα Ανεξαρτησίας, Σύνολα Κάλυψης, και Χρωματικός αριθμός. Ανάκτηση Ιανουάριος 8, 2020, από https://www.softlab.ntua.gr/~fotakis/discrete_math_aegean/indepedence.pdf

Φωτάκης, Δ., & Σπυράκης, Π. (2001). Αλγόριθμοι και Πολυπλοκότητα (Τόμ. Α'). Πάτρα: Ελληνικό Ανοικτό Πανεπιστήμιο.

Χριστοπούλος, Κ. (2015). Ανάπτυξη εφαρμογής σε iphone για την προσομοίωση κβαντικών υπολογισμών. Πάτρα: Πανεπιστήμιο Πατρών/ΗΜ&ΤΥ.