Portfolio Asset Identification using Graph Algorithms on a Quantum Annealer

Angad Kalra
angadk@cs.toronto.edu

Faisal Qureshi
fiqureshi1@cs.toronto.edu

Michael Tisi
michael.tisi@mail.utoronto.ca

Abstract

Our dual objectives are to explore how commercially available quantum hardware and algorithms can solve real world problems in finance, and then to compare quantum solutions to their classical counterparts. Specifically, we use the D-Wave quantum annealing computer (D-Wave 2000Q) to address the problem of asset correlation identification for financial portfolio management. Graphical models offer a natural framework to represent asset correlations. Graphs also naturally map to the quantum annealing hardware architecture developed by D-Wave. We explore how graph algorithms can be implemented on the D-Wave 2000Q machine to cluster asset correlations in order to identify various financial portfolios. Numerical experiments are conducted using four quantum/classical algorithm pairs on four real world financial time series data sets spanning 10 years. For the specific algorithms and datasets selected, the quantum solution is competitive with (and sometimes better than) the classical one. However, quantum fails to scale beyond certain levels of data dimensionality. Our study focuses on comparison of solution quality not speedup. Our results suggest specific high-potential directions for future research.

1 Introduction

Background While the dream of a universal quantum computer may be many years away, we are beginning to witness the commercial availability of limited scope and noisy quantum offerings - the NISQ machines described by Preskill [Preskill, 2018]. One such offering is D-Wave's quantum annealing based 2000 qubit machine: the D-Wave 2000Q. The D-Wave approach has garnered both academic skepticism [Boixo, 2013], [Boixo et al., 2014], [Sco] and results showing potential (some of which are described below). We do not undertake a general evaluation of the performance of the D-Wave computer. Instead, we focus on evaluating whether D-Wave's quantum computer can be used to solve practical problems in finance using real world financial data sets. We then compare quantum results to the results available from "out-of-the-box" classical algorithms run on a modern laptop. [Oce]

Problem We address the problem of identifying assets for various types of portfolio strategies. An important aspect of financial portfolio management is the identification of correlated and uncorrelated assets from a universe of assets. This type of correlation clustering allows the portfolio manager to pick assets in order to implement different diversification and hedging strategies. As described below, the D-Wave machine is limited to solving very specific optimization problems. The challenge is to find problems that can be mapped onto the objective functions supported by the D-Wave architecture. Fortunately, D-Wave is intrinsically suited for certain graph problems, and we exploit this feature by mapping problems of financial asset identification to graph theory problems which can be solved by D-Wave.

Methodology We define time-indexed correlation graphs using various thresholds to model the interrelationships of asset prices over time. We also describe the application of four graph algorithms to identify assets for several portfolio management strategies. We then conduct numerical experiments to compare the performance of four quantum algorithms against their classical counterparts on three financial data sets spanning 10 years. We have not attempted to compare quantum and classical algorithm speeds. In stead our experiments are designed to probe the abilities of quantum algorithms to perform a computationally useful task, and then to evaluate the quality of quantum solutions to the quality of classical algorithm solutions.

Related Work Graph applications are one of the popular topics in D-Wave related research. The company provides extensive referenced documentation and software code accessible here [D-W]. White papers by D-Wave describe programming the graph coloring problem [Dahl] and embedding graphs onto D-wave architecture [dwa]. Using D-Wave for graph similarity problems is discussed in [Hernandez et al.]. Graph partitioning on D-Wave is studied in [Ushijima-Mwesigwa et al., 2017]. Graphs are used for in a quantum-classical hybird approach to cluster analysis in [Neukart et al., 2018]. Structured balance in signed networks is applied to the problem of analyzing social conflicts using the D-Wave machine in [Ambrosiano et al., 2017]. However, to the best of our knowledge, the present study is the first to apply graph algorithms on D-Wave to the domain of financial portfolio management. We have also relied extensively on literature on quantum annealing and adiabetic quantum computing and graph theory, problems and algorithms. References to specific sources is made where the topic is discussed.

Contribution The first contribution of this study is to demonstrate how four graph algorithms (maximum clique, maximum independent set, minimum graph colouring, and signed graph balancing) can be used to identify assets for specific investment strategies. The second contribution of this study is to compare solution quality between D-Wave's quantum computer and popular programming libraries.

Study Outline Section 2 provides background on quantum annealing and D-Wave's approach to it. Section 3 described how graph theory can be applied to asset identification for several investment strategies Section 4 describes our experimental setup and procedure. Section 5 describes our results.

2 Quantum Annealing and D-Wave

In this section we describe D-Wave System's approach to quantum annealing as implemented in the D-Wave 2000Q. Unless otherwise referenced, information regarding D-Wave's annealing technology is obtained from technical documentation provided by the company [D-W].

2.1 D-Wave Systems

D-Wave Systems is a quantum computing company based in Burnaby, British Columbia, Canada. D-Wave's approach to quantum computing is quantum annealing, which is a form of adiabatic quantum computation. D-Wave's latest computer, the D-Wave 2000Q, uses a quantum processing unit (QPU) consisting of 2048 qubits, 6016 couplers, and 128,000 Josephson junctions and operates at temperatures below 15 mK (-273.135^{o}C). Qubits are physically implemented as superconducting loops on the QPU, and are encoded by the direction of current flowing through the loop. Clockwise current flow encodes a $|0\rangle$ qubit and counter-clockwise current flow encodes a $|1\rangle$ qubit. When current flows in both directions simultaneously, the qubit is in superposition. On the QPU, qubits are arranged in *unit cells*, which can have a *column* or *cross* configuration, as shown in Figure 7 of Appendix A. Entanglement between qubits is achieved through the use of couplers, which also serve to interconnect unit cells into an architecture known as the *Chimera* (see Figure 7 in Appendix A). The significance of unit cell configurations and the Chimera architecture is discussed below.

The quantum computers manufactured by D-Wave Systems are by no means universal since they can only solve problems mapped to a specific objective function. In particular, problems must be mapped to conform to an Ising model or Quadratic Unconstrained Binary Optimization (QUBO) function.

2.2 Adiabatic Quantum Computing

Quantum annealing is a form of adiabatic quantum computation which follows the adiabatic quantum theorem:

Adiabatic Quantum Theorem: Let $s := t/T$, $s \in [0, 1]$, and let $H(s)$ be a Hermitian operator that varies smoothly with s. Then, for arbitrarily large T, $H(s)$ varies slowly with t. [Childs, 2008]

The quantum annealing process used by D-Wave can be depicted as an energy diagram (see Figure 8 in Appendix A). Initially, some qubit state is prepared in superposition, with a single energy minimum (a). As the quantum annealing process runs, the system evolves adiabatically in accordance with the time-dependent Schrödinger equation, and a double-well potential is formed, with each minimum representing the energy of qubits in the $|0\rangle$ or $|1\rangle$ state (b). Biases can be introduced to the system to increase the probability of measuring a 0 or 1, which is done by introducing a magnetic field of particular magnitude and direction (c). Once the annealing process is finished and a measurement is made, the system collapses to a classical 0 or 1 state, with lower energies indicating a higher probability of measurement outcome.

D-Wave machines use quantum annealing to find the lowest energy eigenvalue of the Ising Hamiltonian, $\mathcal{H}_{\text{Ising}}$:

$$\mathcal{H}_{\text{Ising}} = -\frac{A(s)}{2}\left(\sum_i \sigma_x^{(i)}\right) + \frac{B(s)}{2}\left(\sum_i h_i\sigma_z^i + \sum_{i>j} J_{i,j}\sigma_z^{(i)}\sigma_z^{(j)}\right) \tag{1}$$

where the first (negative) term represents the (initial) *tunneling Hamiltonian* and the second term represents the final *problem Hamiltonian*. Here, $A(s)$ is the tunneling energy, $B(s)$ is the problem Hamiltonian energy, $\sigma_{x,z}^{(i)}$ are the Pauli matrices acting on qubit i, h_i is the qubit bias, and $J_{i,j}$ is the coupling strength between qubits i and j. At the start of the annealing process, the system is in the energy eigenstate of the tunneling Hamiltonian (i.e., the ground state), and at the end of the process the system's energy eigenstate is that of the problem Hamiltonian.If the annealing process is successful, the system remains in the ground state throughout or finishes in some other local minimum energy state.

2.3 Optimization Models

Ising and QUBO Objective Function For a problem to be solved by D-Wave, it must be mapped onto an Ising or QUBO objective function, which are defined respectively, as follows [Calude, 2017]:

$$E_{\text{Ising}}(\overrightarrow{s}) = \sum_{i=1}^{N} h_i s_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{i,j}s_i s_j \tag{2}$$

$$f(x) = \sum_i Q_{i,i}x_i + \sum_{i<j} Q_{i,j}x_i x_j = \min_{x\in\{0,1\}^n} x^T Q x \tag{3}$$

In the Ising function, h_i and $J_{i,j}$ are the same parameters that appear in $\mathcal{H}_{\text{Ising}}$, and in the QUBO function, Q is an upper-triangular $N \times N$ matrix of real coefficients which serve as weights.[Calude, 2017] Note that both equations take binary inputs (i.e. $s \in \{-1, 1\}$ and $x \in \{0, 1\}$) which can be converted easily by the formula $s = 2x - 1$.

Minor-Embedding Graph Problems onto the Chimera Architecture Once a problem is mapped to an Ising or QUBO equation, one further action is required before it can be solved by D-Wave. From the objective function, the variables must be mapped to the qubits; a process known as *minor-embedding*. The first step is to convert an Ising model or QUBO function into a graph. When mapping a function to a graph, each node represents the bias of a binary variable, and each edge represents the coupling strength between two variables. The graph is then mapped to the Chimera architecture via the minor embedding process where the number of qubits used on the QPU must be greater than or equal to the number of binary variables present in the Ising/QUBO function. This process is illustrated in Figure 9 of Appendix A.

3 Graph Algorithms for Financial Portfolio Management

In the previous section, we describe how the D-Wave machine uses quantum annealing to solve optimization problems. In this section, we demonstrate how graph theory can be applied to the problem of asset identification for portfolio management. This enables us to solve such problems on the D-Wave quantum annealer.

3.1 Diversification and Hedging in Financial Portfolios

The identification of correlated and uncorrelated securities is an important aspect of financial portfolio selection. Diversification and hedging are two widely used strategies to manage the risk of a financial portfolio. Diversification seeks to reduce the volatility of returns of a portfolio by incorporating uncorrelated assets into the portfolio. Under this strategy, the portfolio manager seeks to identify uncorrelated assets for inclusion in a portfolio while removing assets that may have become correlated with other assets in the portfolio. Hedging is a strategy to reduce risk for a given position. For example, if a portfolio manager acquires asset A on the belief that its price will rise, she may also want to "hedge" her risk by acquiring (a smaller position in) asset B whose price is negatively correlated to the price of asset A. If A goes up, she stands to make a profit from A and take a small loss from B. If, on the other hand A falls (i.e. she was wrong), then her loss from A is reduced by a small profit from B. By hedging, the portfolio manager has reduced her expected risk (as well as return). Hedging can also apply to baskets of assets, where one basket is used to hedge the other basket. In this case, each basket should contain positively correlated assets, but any two assets picked from different baskets should be negatively correlated with each other. Thus the diversification strategy seeks uncorrelated assets, while the hedging strategy seeks positively and negatively correlated assets.

3.2 Correlation Graphs for Financial Assets

Graph Terminology A graph $\boldsymbol{\mathcal{G} = (\mathcal{V}, \mathcal{E})}$, consists of $\boldsymbol{\mathcal{V} = \{v_1, ...v_k\}}$, a finite set of vertices (also known as nodes), and $\boldsymbol{\mathcal{E} \subseteq V \times V}$, a finite set of edges connecting (some or all of) the vertices. An edge connects two vertices of $\boldsymbol{\mathcal{G}}$ and therefore the two vertices are considered adjacent if and only if $\boldsymbol{(v_i, v_j) \in \mathcal{E}(\mathcal{G})}$. Edges are graphically depicted as lines connecting graph vertices. A subgraph $\boldsymbol{\mathcal{G}_1}$ of $\boldsymbol{\mathcal{G}}$ consists of a set of vertices $\boldsymbol{\mathcal{V}_1 \subseteq \mathcal{V}}$, and the subset of edges $\boldsymbol{\mathcal{E}_1 \subseteq \mathcal{E}}$ connecting $\boldsymbol{\mathcal{V}_1}$. Furthermore, if $\boldsymbol{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3}$ are subgraphs of $\boldsymbol{\mathcal{G}}$, then $\boldsymbol{\mathcal{G}_2}$ is said to separate $\boldsymbol{\mathcal{G}_1}$and $\boldsymbol{\mathcal{G}_3}$ if every path between $\boldsymbol{\mathcal{G}_1}$and $\boldsymbol{\mathcal{G}_3}$ intersects a node in $\boldsymbol{\mathcal{G}_2}$. A graph (subgraph) is complete, if every pair of vertices in the graph (subgraph) are connected by an edge. A complete subgraph of a graph $\boldsymbol{\mathcal{G}}$, is also called clique of $\boldsymbol{\mathcal{G}}$. The cardinality of a graph (subgraph) is defined as the number of vertices in the graph (subgraph). An independent set in a graph is defined as a set of vertices such that no two vertices from this set is connected by an edge. An independent set is also known as an anti-clique.

Time Indexed Correlation Graphs In identifying assets for diversification and hedging, we are interested in understanding how prices of assets vary over time, individually and relative to each other. Graphical models offer an intuitive and robust framework for capturing and analyzing multivariate relationships. Temporal extensions to graphical models facilitate the analysis of multivariate relationships over time (see Related Work, Section 1).

In this study, we shall work with correlation graphs indexed over time, where the vertices of the graphs represent assets and the edges represent the relationships between the assets. Specifically, assume each dataset $\mathcal{D}$ consists of a set of K assets $\mathcal{A} = \{a_k\}$ for $k \in \{1, ..K\}$, and corresponding daily prices of these assets, $price({a_k}^{(t)})$ for $t \in daterange[T]$ for some time period T. To model how prices change over time, we normalize $price({a_k}^{(t)})$ into into a daily price return ${x_k}^{(t)}$ as follows:

$$Daily\ Price\ Return = {x_k}^{(t)} = \frac{price({a_k}^{(t)}) - price({a_k}^{(t-1)})}{price({a_k}^{(t-1)})} \tag{4}$$

To capture multivariate covariances, we require multiple samples of data. Since we have daily price returns, we must select a time-window over which to measure inter-asset covariances. As is standard practice in finance, we use a calender month for calculating asset covariances. Assuming $\boldsymbol{x^{(t)}} = (x_1^{(t)}, x_2^{(t)}, ...x_K^{(t)})$, and the sample mean $\bar{\boldsymbol{x}}$ is known, we construct a monthly covariance and correlation matrices $\boldsymbol{S_m}$ and $\boldsymbol{C_m}$, respectively, as follows:

$$\begin{gathered} \textit{Sample Covariance Matrix} = \boldsymbol{S_m} = 1/N \sum_t (\boldsymbol{x^{(t-i)}} - \bar{\boldsymbol{x}})(\boldsymbol{x^{(t-i)}} - \bar{\boldsymbol{x}})^T \\ \text{where } N = NumberOfDaysIn[month\ m] \\ \text{and } t \in DateRange[month\ m] \end{gathered} \tag{5}$$

$$\textit{Sample Correlation Matrix} = \boldsymbol{C_m} = (diag(\boldsymbol{S_m}))^{\frac{1}{2}} (diag(\boldsymbol{S_m}))^{\frac{1}{2}} \tag{6}$$

Each monthly graph $\boldsymbol{\mathcal{G}_m}$ is then constructed such that vertex set $\{v_k\}$ corresponds to the asset set $\{a_k\}$, and the edges correspond to the entries in the correlation matrix $\boldsymbol{C_m}$. Because we are dealing with continuous data, we will generally have a non-zero correlation for each asset pair, and thus a complete graph. In this study, we will characterize asset pairs as being either correlated or uncorrelated based on whether or not the correlation between the pair exceeds a minimum level. We introduce a correlation threshold level parameter $\lambda \in [0, 1]$ to create sparsity in the graphs, as follows:

$$\begin{aligned} corr(a_i, a_j) < \lambda, \longrightarrow (v_i, v_j) \notin \mathcal{E}(\mathcal{G}_m) \\ corr(a_i, a_j) => \lambda, \longrightarrow (v_i, v_j) \in \mathcal{E}(\mathcal{G}_m) \end{aligned} \tag{7}$$

This will result in a set of thresholded time-indexed correlation graphs $\boldsymbol{\mathcal{G}_m^{\lambda}}$. The thresholding level of graph is inversely related to its density, defined as:

$$Density(\mathcal{G}) = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)} \tag{8}$$

3.3 Graph Algorithms

3.3.1 Maximum Clique and Maximum Independent Set

Definition The maximum clique problem (MCP) is to find the clique with maximum cardinality in a given graph. MCP is one of the most widely studied graph problems because of the diversity of possible applications. MCP is famously included in Karp's list of 21 NP-complete problems [Wik, b], and is fixed parameter intractable and difficult to approximate [Wik, c]. The maximum independent set problem (MISP) is to find the largest independent set in a graph (i.e. the largest set of vertices such that no two vertices in this set are connected by an edge)[Wik, d]. MCP is strictly equivalent to MISP: the maximum clique of a graph $\boldsymbol{\mathcal{G}}$ is equal to the maximum independent set of the graph's complement $\bar{\boldsymbol{\mathcal{G}}}$ [Wu and Hao, 2015].

Application As discussed in Section 1, MCP has been applied to problems in a wide range of domains including bioinformatics, chemoinformarics, economics, scheduling, social networks, economics and finance. In this study we propose that given a set of assets, MCP can be used to identify the largest subset of correlated assets. This is useful for making a concentrated bet in one direction (i.e. by taking positions in only positively correlated assets) or for hedging (by taking positions in negatively correlated assets). Analogously, we also propose that MISP can be used to identify the largest subset of uncorrelated assets in a set of assets, which is valuable for creating a diversified portfolio.

Classical Algorithms Classical algorithms to solve MCP and MISP are generally divided into exact approaches that employ some variation of the branch-and-bound method and heuristic approaches that include employ greedy search, localized search or genetic methods. A detailed discussion of the various classical algorithms proposed so solve MCP and MISP is outside the scope of this paper. The interested reader is referred to the most recent literature review from 2015 [Wu and Hao, 2015], and the most major prior one from 1999 [Bomze et al., 1999].

3.3.2 Minimum Graph Colouring

Definition The Graph Colouring Problem (GCP) is to label (colour) the vertices of graph such that no two vertices connected by an edge have the same label (colour). The Minimum Graph Colouring Problem (MGCP) is to find for a given graph, the graph colour with the least number of labels (colours). Other graph colouring problems such as edge colouring and face colouring can be transformed to MGCP. The smallest number of colors required to colour a graph is called its chromatic number. The problem of determining the chromatic number of a graph is NP-hard [Erdos, 1959].

Application MGCP has been applied to scheduling, planning and pattern matching problems in many domains [Wik, a]. For portfolio management, graph colours can be seen as baskets of assets. Thus the solution to the MGCP for a given correlation graph of assets will provide the minimum number of baskets of assets such that each basket contains uncorrelated assets. In other words, MGCP yields the minimum number of diversified portfolios you can construct from a set of assets. Conversely, the solution to MGCP for the complement of a correlation graph yields the minimum set of baskets such that each basket contains correlated assets. In this case, each basket can for example be used to execute a concentrated position or hedging strategy by picking only positively correlated assets, or negatively correlated asset pairs, respectively. To the best of our knowledge, this is a novel application of MGCP.

Classical Algorithms Major classical approaches to solving MGCP include greedy colouring, subgraph expansion and recursive methods. A description of classical algorithms to solve MGCP is outside our scope but we refer the interested reader to a comprehensive review here [Husfeldt, 2015].

3.3.3 Structural Balance

Definition A signed graph is a graph where edges are labelled as being either positive or negative. A signed graph is defined to be perfectly balanced when its vertices can be divided into two clusters such that any two vertices within a cluster are connected by a positive edge, and any two vertices from different clusters are connected by a negative edge. A graph is defined as imbalanced if no clusters can be defined that make it balanced - in this case, the graph's frustration index is computed as the minimum number of edges that need to be deleted to make it perfectly balanced. [Aref, 2017]. The Structural Balance Problem (SBP) is to calculate the frustration index of a signed graph, and is known to be a NP-hard problem [Hüffner et al., 2010]. SBP can be generalized to a multicluster context resulting in the Correlation Clustering Problem [Levorato et al., 2017].

Application The concept of structural balance in a signed graph was proposed by [Harary et al., 1953] in 1953. Since then SCP has been applied to numerous domains including traditional and online social networks, biological networks, international relations, chemistry, physics. SCP has been applied to financial portfolios by [Harary et al., 2002], who suggest the use of structurally balanced graphs for constructing hedging portfolios. We have followed this approach. Determining the most balanced signed graph yields two baskets such that positive correlation is maximized (and negative correlation minimized) within each basket, and negative correlation is maximized (and positive correlation minimized) between baskets. This asset allocation is useful for a hedging strategy.

Classical Algorithms A description of classical algorithms to solve SBP is outside our scope but we refer the reader to [Aref, 2017] and [Levorato et al., 2017].

3.3.4 Ising/QUBO formulations

The four graph problems can be mapped onto the D-Wave objective functions in order to solve via quantum annealing. Given graph $\boldsymbol{\mathcal{G} = (\mathcal{V}, \mathcal{E})}$, MISP and MGCP map to QUBO formulation while SBP maps to an Ising formulation. (MCP can be solved by solving for MISP on $\bar{\boldsymbol{\mathcal{G}}}$.) The coefficient matrices for MISP and SBP are provided below. The MGCP formulation is more involved, and can be found here [Dahl].

$$Q_{i,j}^{MISP} = \begin{cases} 1 & \text{if } (i,j) \in \boldsymbol{\mathcal{E}}, \text{ and } i < j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$J_{i,j}^{SBP} = \begin{cases} sign(i,j) & for (i,j) \in \boldsymbol{\mathcal{E}}, \text{ and } i < j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

4 Numerical Experiments

The dual objectives of this study were to explore how commercially available quantum annealing technology (i.e. D-Wave 2000Q) can be used to solve real world problems, and then to compare these quantum solutions to classical ones. In previous sections we described how quantum graph algorithms can be applied to the domain of financial portfolio management. In this section we describe numerical experiments conducted to compare the performance of commercially available quantum and classical solutions for our selected financial problems.

4.1 Setup and Procedure

Computing Resources We used Ocean Software [Oce], a python based library offered by D-Wave to interact with the D-Wave 2000Q machine (described above) through the online Leap interface system [DLe] recently made available by the company. Classical algorithms were run on a 2015 Macbook Pro with Intel i5 2.7 GHz CPU and 8 GB RAM.

Graph Algorithms Quantum annealing based graph algorithms provided in the Ocean Software Dwave_NetworkX library were executed with default parameters without performance tuning. Graph data structures were implemented in the popular Python NetworkX graph library [PyN]. Classical graph algorithms corresponding to the above quantum algorithms were also selected from the NetworkX library. A classical version of the structural balance algorithm was not included in the scope of the current study due to unavailability of an appropriate option. Quantum and classical algorithms are listed in Table 1.

Data Four financial datasets (Asset Class, Sectors, SP, FTSE) were prepared in the experiments. These are described in Table 2. Data was sourced from the Center for Research in Security Prices [CRS]. For each dataset, 2519 days (10 years) of daily security prices were retrieved. Time indexed correlation graphs were generated for each of the datasets (as described in 3.2, such that each graph represented one calender month of data.

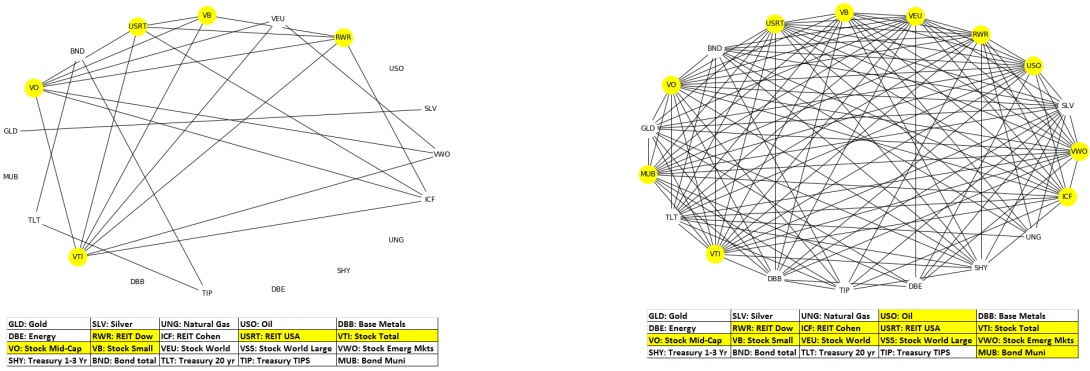
Experimental Procedure Monthly correlation graphs were constructed for the four datasets, and thresholding was applied. The set of thresholded monthly correlation graphs is described in Table 3. D-Wave failed to execute the quantum algorithms on the SP dataset, presumably due to the significantly higher dimensionality of this dataset. Consequently we did not record classical algorithm results for this dataset as well. Also, as mentioned previously we did not run a classical version the Structural Balance algorithm.

5 Results

5.1 Quantum Solutions to Graph Problems

The first objective of this study was to apply quantum algorithms to the real world problem of identifying assets for various financial portfolio strategies. In this section we describe some sample solutions yielded by the quantum graph algorithms.

Maximum Clique

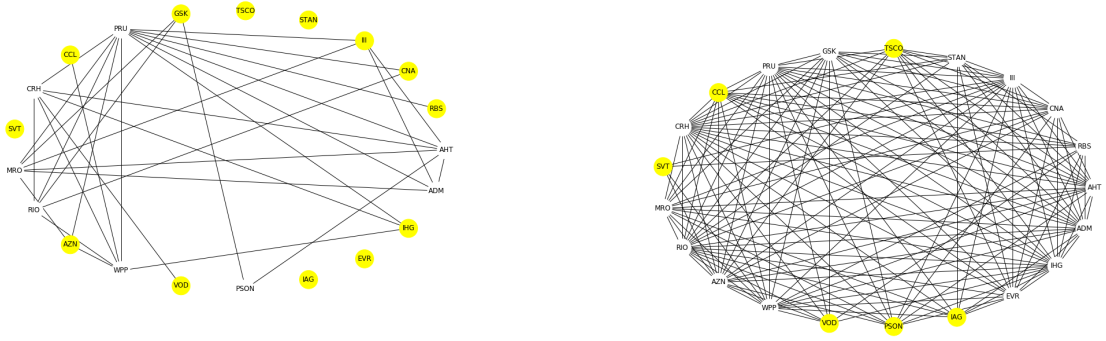


(a) Asset Class, Low Density

(b) Asset Class, High Density

Figure 1: Quantum MCP solutions for the Asset Class dataset are plotted. At low density the identified maximum clique consists only of the US based stock and real estate assets, which are known to be tightly correlated. At high density the identified maximum clique includes all asset classes except commodity and treasuries. Commodities have a physical supply-demand dynamic separate from financial markets, and the price of treasuries is tied to confidence in the US government.

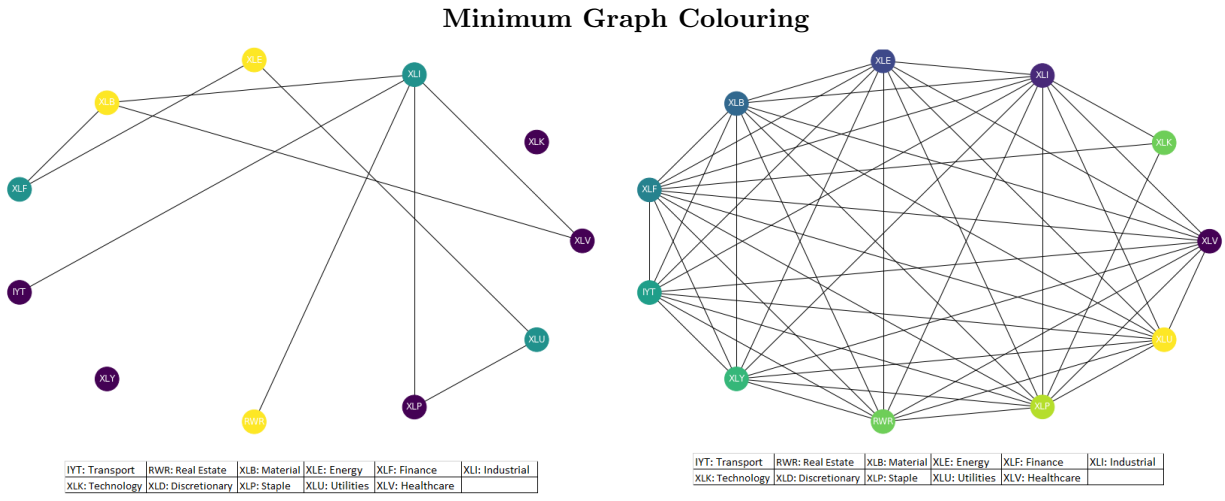
Maximum Independent Set



(a) FTSE, Low Density

(b) FTSE, High Density

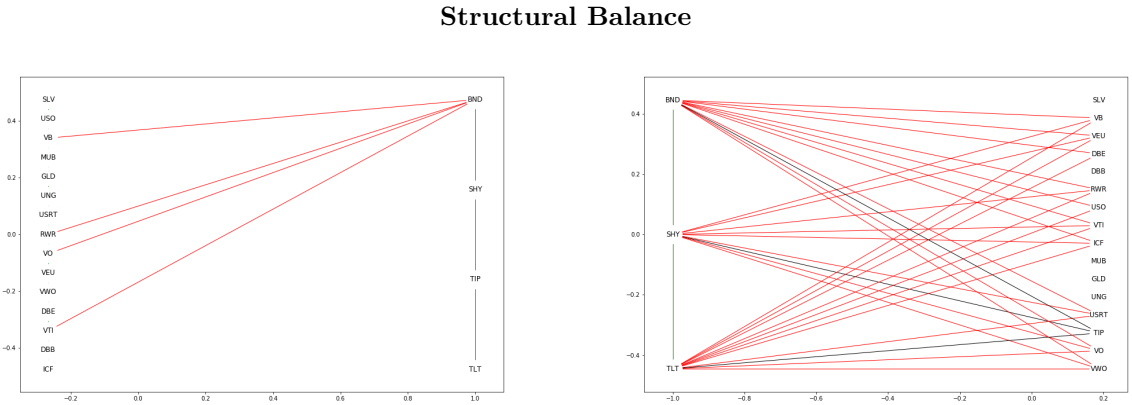
Figure 2: Quantum MISP solutions for the FTSE dataset are plotted. At low density, 12 stocks (the maximum independent set from amongst 19) are identified as the largest possible diversified portfolio. At high density, the identified largest possible diversified portfolio is smaller, consisting of only 6 stocks.



(a) Low Density

(b) High Density

Figure 3: Quantum MGCP solutions for the Sectors dataset are plotted. At low density, 3 diversified baskets are identified: Yellow (real estate, energy, materials), Purple (consumer discretionary, consumer staple, healthcare, technology), Blue (utilities, industries, finance). At high density, no perfectly diversified portfolio can be created as every node is coloured differently.



(a) Low Density

(b) High Density

Figure 4: Quantum SBP solutions for the Asset Class dataset are plotted. Red/blue edges represent negative/positive correlation, and black edges represent frustration. At low density, the graph is perfectly balanced into two clusters, where each cluster contains positively correlated assets only. Four of the assets in the left cluster (VB, VTO, VI are negatively correlated to bonds(BND) from the right cluster. These could be paired for a hedging strategy. At high density, a perfectly balanced network is not possible.

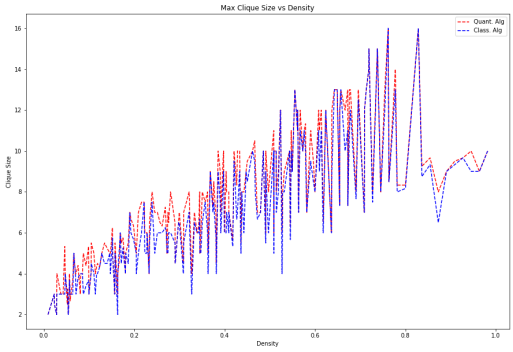
5.2 Comparison: Quantum vs. Classical

As discussed, our focus is the quality of algorithm solutions not algorithm speed-ups. The quality of solution is measured by a solution score defined in a straight-forward manner: For MCP and MISP, the solution score is the cardinality of the returned maximum clique and maximum independent set, respectively. For MGCP, the quality score is the minimum number of colours returned. For SBP, the quality score is the number of frustrated edges (i.e. the frustration index). Thus for MCP and MISP, a bigger score is better, while for MGCP and SBP a smaller score is better. In Appendices C through F, we have plotted for each algorithm

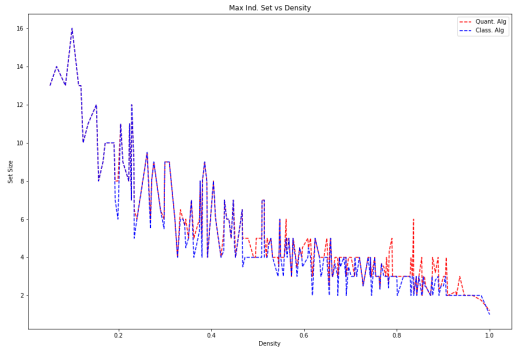
the quantum and classical score vs. graph density for various thresholded graphs. (In the case of SBP, only quantum scores are displayed).

Below we summarize solution scores for the four algorithms across all datasets and graph densities (see Figure 5). We can make several observations. Predictably, as density increases, MCP and MGCP scores increase because it the maximum clique size and minimal graph colouring number are both expected to be larger for more dense graphs. Conversely, as density increases, MISP scores decrease because independent sets generally become smaller as a graph becomes more dense. It is interesting to note that quantum seems to perform better on MCP at all densities, and better on MISP at high densities. At MGCP, quantum and classical algorithms seem to perform competitively. Though we don't have a classical comparison for SBP, we note an interesting pattern in the frustration score. As expected, there is low frustration at low densities (it is easier to balance a sparser network). However, at higher densities, we notice an interesting phenomenon. Firstly, the frustration seems to maximize at .8 density and fall thereafter. We do not understand why frustration goes down even as density increases beyond this point. Secondly, unlike the other algorithms, perfect SBP scores are interspersed between poor SBP scores i.e. perfect balance is achieved at densities close to where the worst balance also occurs. This occurs because one of our datasets (Sectors) ad only positively correlated nodes, and therefore the algorithm was able to perfectly balance this dataset for all densities.

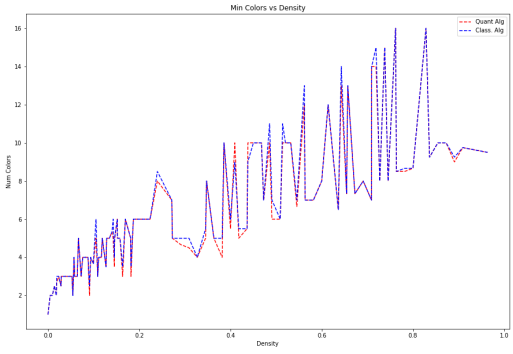
Figure 5: Graph Algorithm Comparison: Score vs. Density



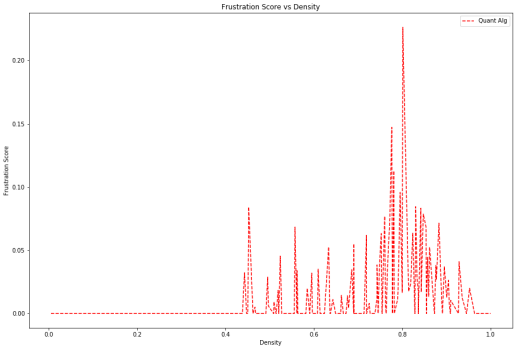
(a) Maximum Clique Number vs. Density



(b) Maximum Independent Set Number vs Density



(a) Minimum Graph Colours vs. Density



(b) Minimum Frustration Score vs. Density

6 Conclusion

We demonstrated that quantum annealing graph algorithms run on D-Wave 2000Q can be used effectively for the real world problem of identifying assets for specific investment strategies. We also discovered that these algorithms do not scale effectively for graphs with more than a few dozen vertices. We found that quantum solutions are marginally superior for MCP and MISP and competitive for MGCP. We also noted some interesting patterns in the quantum algorithm for SBP.

6.1 Future Directions

The present study combined the goal of investigating if graph algorithms on current quantum computers can can solve useful financial problems and then to study the quality of these solutions. These two objectives should be further pursued separately. The former objective should be pursued by investigating how more complex asset identification problems can be mapped onto the Ising/QUBO models of quantum annealing. The latter objective is perhaps even more interesting to pursue as described below.

Our results demonstrated that quantum seems to perform better on MCP at all densities, and better on MISP at high densities. These areas should be studied in depth. A comprehensive range of graph formations should be generated synthetically, and the D-Wave quantum algorithms for MCP and MISP should be compared to state of the art classical algorithms over a range of graph sizes and densities. Another interesting area is to identify state-of-the-art in classical SBP, and then compare it similarly to quantum SBP. The distribution in respective results may yield interesting findings on whether quantum annealing produces quantitatively different results. Such differences could be exploitable for perhaps different problems. Of course, in addition to solution quality, quantum vs classical speedup is also worth investigating.

References

Center for research in security prices. http://www.crsp.com/. Accessed: 2018-11-01.

Getting started with the d-wave system: User manual. https://docs.dwavesys.com/docs/latest/doc_getting_started.html. Accessed: 2018-12-07.

Dwave leap. https://cloud.dwavesys.com/leap/login/?next=/leap/. Accessed: 2018-11-01.

Ocean tools documentation. http://https://docs.ocean.dwavesys.com/en/latest/index.html. Accessed: 2018-11-01.

Python networkx. https://networkx.github.io/. Accessed: 2018-11-01.

Scott aaronson blog. https://www.scottaaronson.com/blog/. Accessed: 2018-12-07.

Graph colouring. https://en.wikipedia.org/wiki/Graph_coloring#Applications, a. Accessed: 2018-11-01.

Karps 21 np-complete problems. https://en.wikipedia.org/wiki/Karp's_21_NP-complete_problems, b. Accessed: 2018-11-01.

Clique problem. https://en.wikipedia.org/wiki/Clique_problem, c. Accessed: 2018-11-01.

Independent set. https://en.wikipedia.org/wiki/Maximal_independent_set, d. Accessed: 2018-11-01.

Virtual Graphs for High Performance Embedded Topologies. Technical report.

John Joseph Ambrosiano, Randy Mark Roberts, and Benjamin Hayden Sims. Using the d-wave 2x quantum computer to explore the formation of global terrorist networks. Technical report, Technical report LA-UR-17-23946, Los Alamos National Laboratory, 2017.

Samin Aref. Balance and frustration in signed networks under different contexts. arXiv preprint arXiv:1712.04628, 2017.

S Boixo. S. boixo, t. albash, fm spedalieri, n. chancellor, and da lidar, nat. commun. 4, 2067 (2013). Nat. Commun., 4:2067, 2013.

Sergio Boixo, Troels F Rønnow, Sergei V Isakov, Zhihui Wang, David Wecker, Daniel A Lidar, John M Martinis, and Matthias Troyer. Evidence for quantum annealing with more than one hundred qubits. Nature Physics, 10(3):218, 2014.

Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. The maximum clique problem. In Handbook of combinatorial optimization, pages 1–74. Springer, 1999.

Hua Calude, Dinneen. Qubo formulations for graph isomorphism problem and related problems. Theoretical Computer Science, 701:54–69, 2017.

Andrew Childs. Lecture 18: The quantum adiabatic theorem. University of Waterloo, pages 1–5, 2008. Accessed: 2018-12-07.

E.D. Dahl. Programming with D-Wave: Map COloring Problem. Technical report.

Paul Erdos. Graph theory and probability. canad. J. Math, 11(11):34–38, 1959.

Frank Harary, Meng-Hiot Lim, and Donald C Wunsch. Signed graphs for portfolio analysis in risk management. IMA Journal of management mathematics, 13(3):201–210, 2002.

Frank Harary et al. On the notion of balance of a signed graph. The Michigan Mathematical Journal, 2(2): 143–146, 1953.

Maritza Hernandez, Arman Zaribafiyan, Maliheh Armamon, and Mohammad Naghibi. Quantum Approaches to Graph Similarity. Technical report.

Falk Hüffner, Nadja Betzler, and Rolf Niedermeier. Separator-based data reduction for signed graph balancing. Journal of combinatorial optimization, 20(4):335–360, 2010.

Thore Husfeldt. Graph colouring algorithms. In Encyclopedia of Mathematics and its Applications, chapter 13, pages 277–303. Cambridge University Press, 2015.

Mario Levorato, Rosa Figueiredo, Yuri Frota, and Lúcia Drummond. Evaluating balancing on social networks through the efficient solution of correlation clustering problems. EURO Journal on Computational Optimization, 5(4):467–498, 2017.

Florian Neukart, David Von Dollen, and Christian Seidel. Quantum-assisted cluster analysis on a quantum annealing device. Frontiers in Physics, 6:55, 2018.

John Preskill. Quantum computing in the nisq era and beyond. arXiv preprint arXiv:1801.00862, 2018.

Hayato Ushijima-Mwesigwa, Christian FA Negre, and Susan M Mniszewski. Graph partitioning using quantum annealing on the d-wave system. In Proceedings of the Second International Workshop on Post Moores Era Supercomputing, pages 22–29. ACM, 2017.

Qinghua Wu and Jin-Kao Hao. A review on algorithms for maximum clique problems. European Journal of Operational Research, 242(3):693–709, 2015.

7 Appendix A: D-Wave Annealing Figures

Figure 7: D-Wave Chimera Architecture

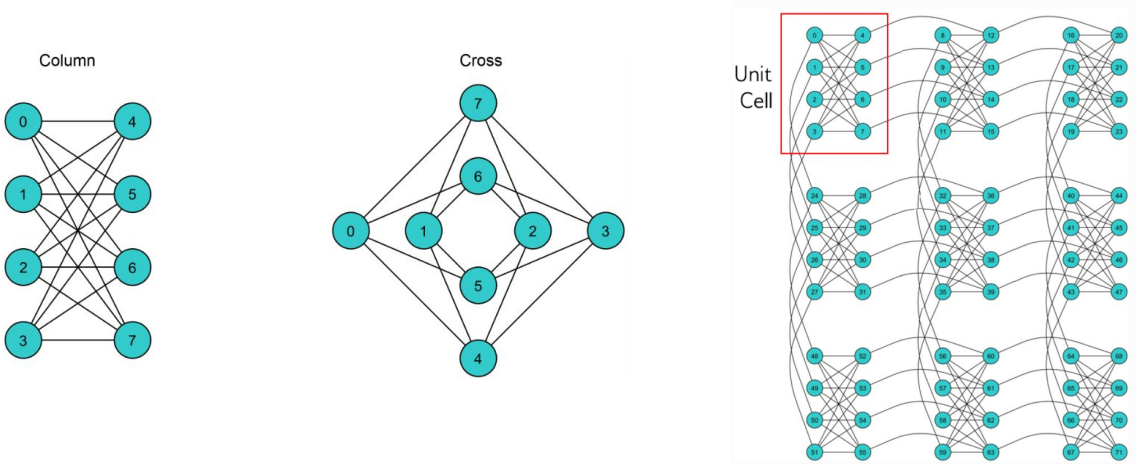


Figure 8: Quantum Annealing Energy Diagram

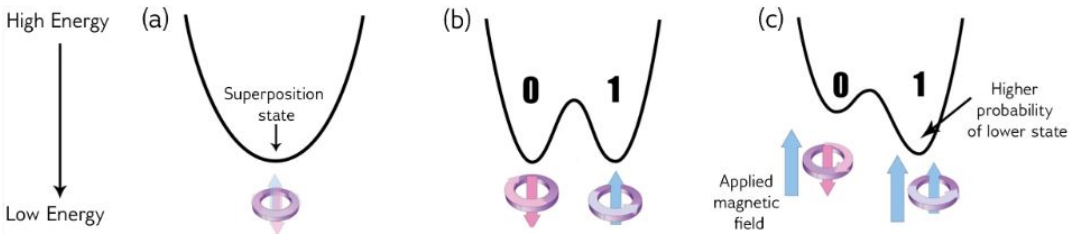
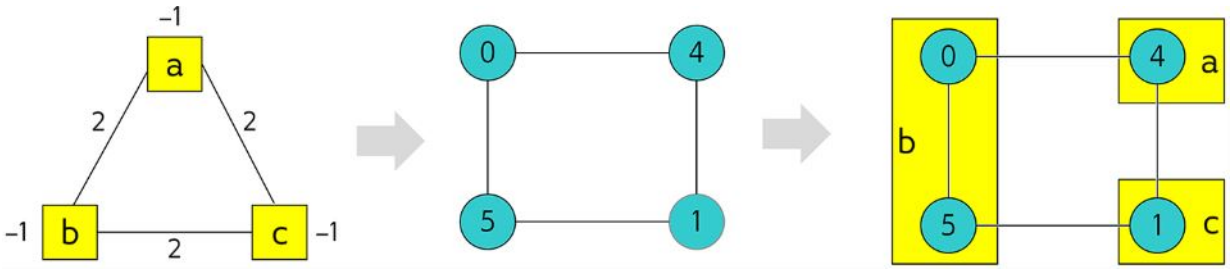


Figure 9: QUBO Mapping and Minor Embedding



The left most figure maps QUBO function $E(x, y, z) = 2xy + 2xz + 2yz - x - y - z + 1$ on to a graph. The next two figures "minor embed" the QUBO onto D-Wave's Chimera Architecture.

8 Appendix B: Experiment Related

Table 1: Graph Algorithms

Quantum (dwave_networkx)	Classical (networkx.algorithms)
maximum_clique()	approximation.clique.max_clique()
maximum_independent_set()	approximation.clique.max_independent_set()
min_vertex_coloring()	coloring.greedy_color()
structural_imbalance()	n/a

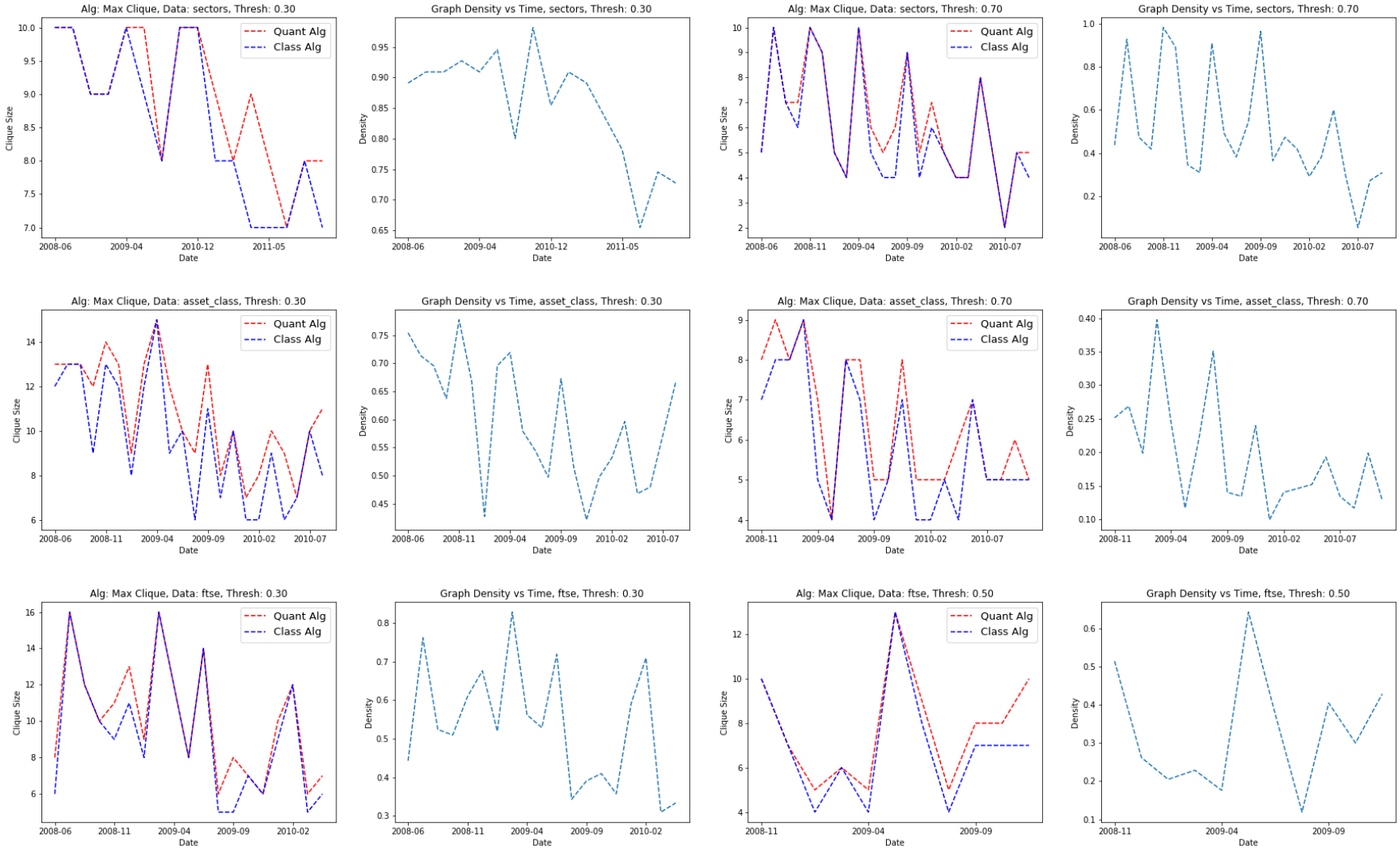
Table 2: Datasets

Datasets	#	Description
Asset Classes	19	major ETFs representing asset classes i.e. equity, debt, RE, commodities etc.
Sectors	11	major ETFs representing sectors i.e. utilities, finance, energy etc.
FTSE	22	stocks from London based FTSE 100 Index
S&P	90	stocks from US based S&P 100 Index

Table 3: Thresholded Graphs

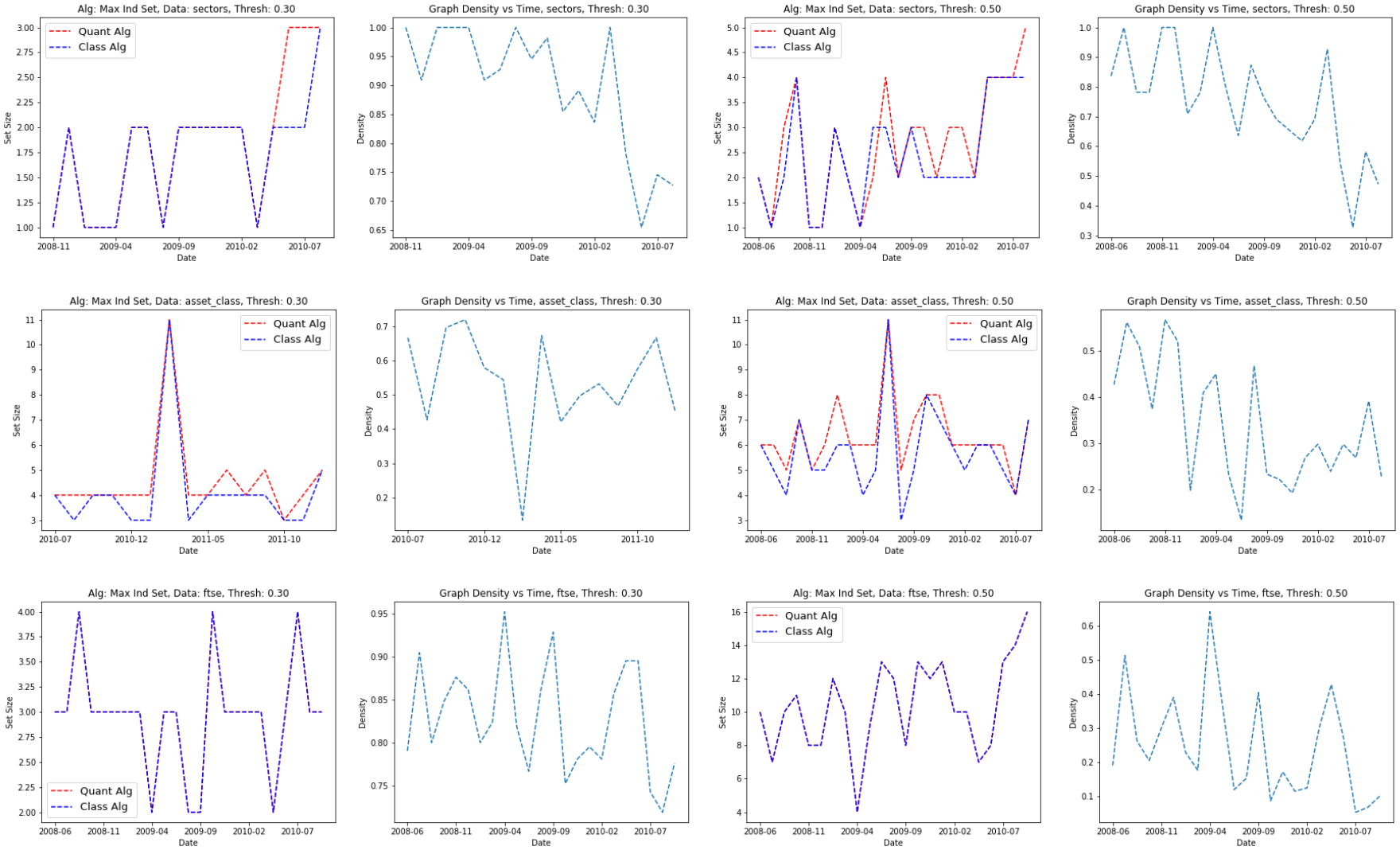
Dataset	Thresh. Levels	# Months	Total Graphs
Asset Classes	.5,.6,.7,.8,.9	120	600
Sectors	.5,.6,.7,.8,.9	120	600
FTSE	.5,.6,.7,.8,.9	120	600
S&P	.5,.6,.7,.8,.9	120	600
			2400

9 Appendix C: Max Clique - Quantum vs. Classical Algorithm Scores



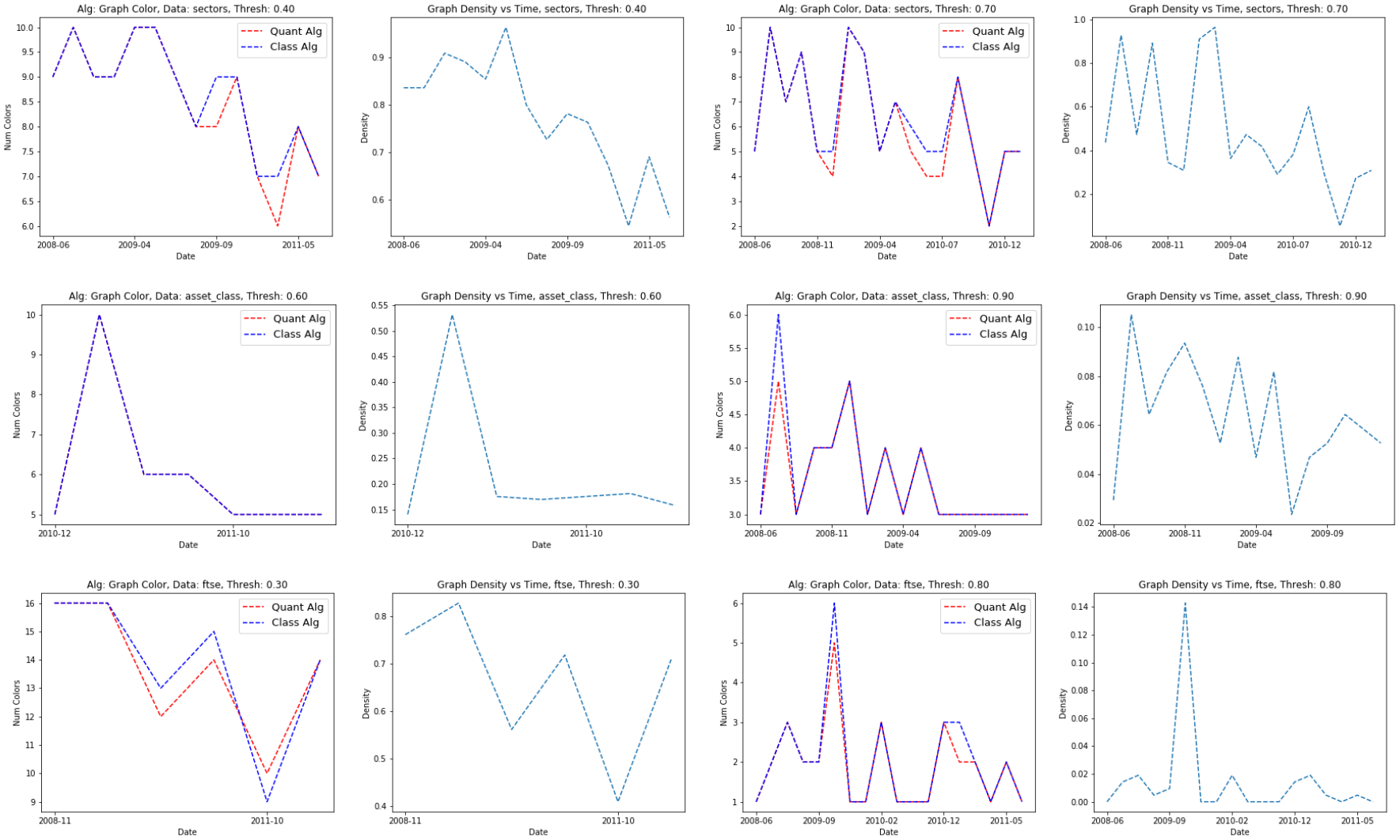
Sizes of max cliques found are plotted alongside density of the underlying graph. For each dataset, high and low density thresholds are displayed.

10 Appendix D: Max Independent Set - Quantum vs. Classical Algorithm Scores



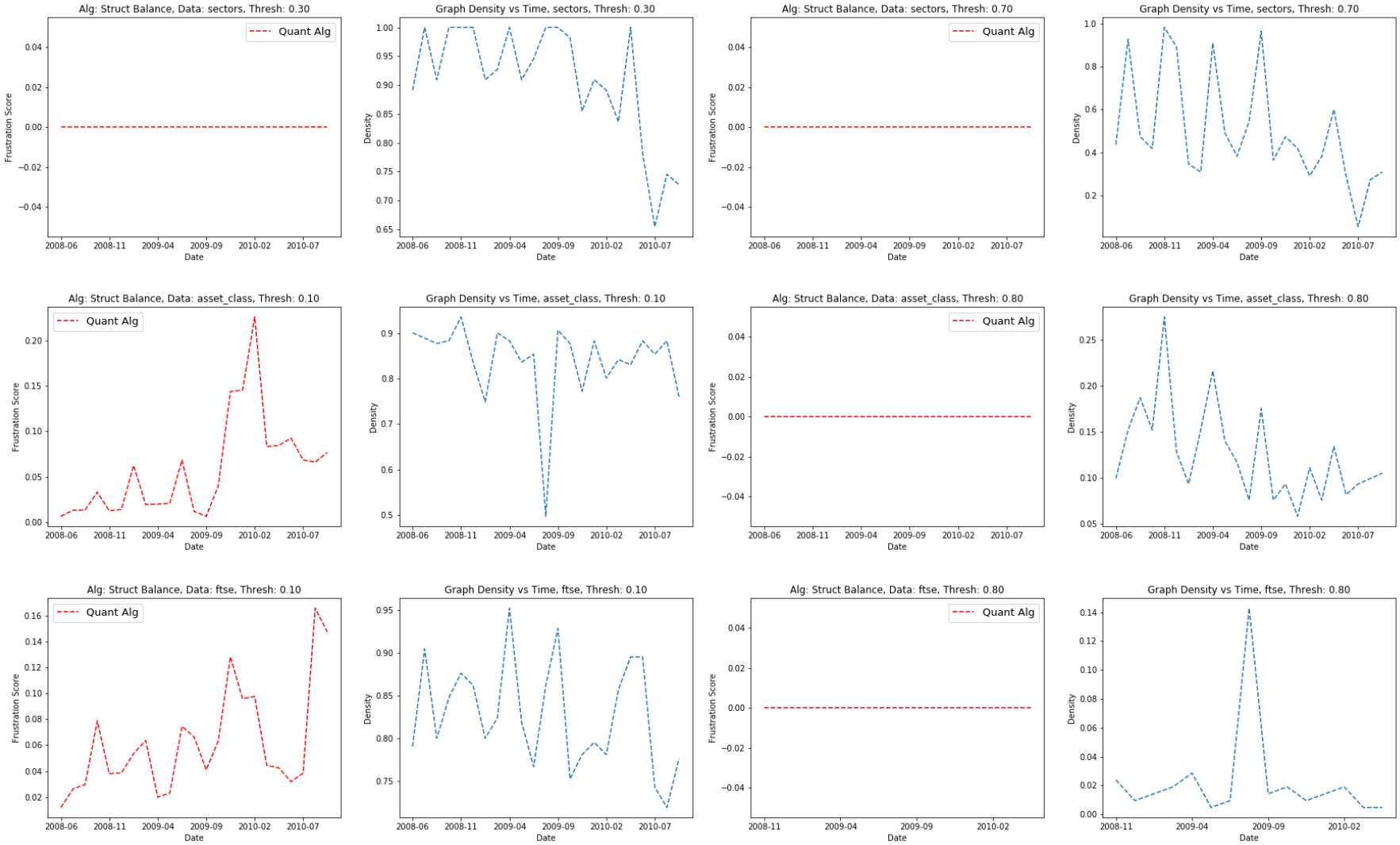
Max indep. set sizes found are plotted alongside density of the underlying graph. For each dataset, high and low density thresholds are displayed.

11 Appendix E: Graph Coloring - Quantum vs. Classical Algorithm Scores



Minimum colourings found are plotted alongside the density of the underlying graph. For each dataset, high and low density thresholds are displayed.

12 Appendix F: Structural Balance - Quantum vs. Classical Algorithm Scores



Frustration Score is plotted alongside the density of the underlying graph. For each dataset, high and low density thresholds are displayed.