A Spectrum Allocation and Auction Mechanism with the Maximum Weight Independent Set

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Master of Science

by

Hui Boon Low

August

2015

APPROVAL SHEET

The thesis

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science

Digitally signed by HuI BOON LOW
DN: cn=HuI BOON LOW, o, ou,
email=ginjoints@gmail.com, c=US
Date: 2015.08.01 11:07:09 +08'00'

AUTHOR

The thesis has been read and approved by the examining committee:

Alfredo Garcia

Advisor

Gerard Learmonth

Zongli Lin

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, Dean, School of Engineering and Applied Science

August

2015

A Dynamic Spectrum Allocation and Auction Mechanism with the Maximum Weighted Independent Set

---

Masters Of Science Thesis

Presented To

The Faculty Of The School Of Engineering And Applied Science

University Of Virginia

In Partial Fulfillment

Of The Requirements Of The Degree

Master of Science in System Engineering

By Hui Boon Low

17 April 2015

1 Acknowledgements

I would like to thank my advisor, Professor Alfredo Garcia, for his constant support and patience. I am also grateful to my thesis committee members, Professor Lin Zongli and Professor Gerard Learmonth for their time and guidance. Thank you, also, to the faculty, staff, and students of the Systems and Information Engineering Department at University of Virginia for helping me make my experience here unforgettable. I wouldn't have made it through without the friendships with some of my classmates.

Finally, my utmost appreciation goes to my family members for their unwavering support and understanding.

Contents

2 Abstract

The problem of spectrum allocation has become more pertinent in recent times, especially due to the rapid increase in demand for mobile data. The majority of spectrum now is being statically allocated according to usage functions. For example, 30-300MHz is reserved partly for radio and TV broadcasts, and 300-3000MHz is reserved partly for mobile phone usage and wireless LAN. Hence, there is high variance in the overall spectrum utilization, with certain frequencies being underutilized and others facing spectrum shortages. In order to improve utilization rates, dynamic spectrum access (DSA) methods are being used to match demand for spectrum with un-utilized spectrum dynamically, using cognitive radio technology. DSA methods are computationally efficient to ensure that spectrum allocation to users is updated on a timely basis, to maximize utilization rates.

The reusability property of radio spectrum allows a single channel to be allocated to multiple users at the same time, subject to interference constraints. In this thesis, we look at the scenario where the utility that different users derive from spectrum usage vary. The utility of each user is expressed by the user in the form of an auction bid, of which the truthfulness is ensured by the VCG pricing mechanism.

The allocation of radio spectrum to maximize social efficiency, subject to interference constraint, is the Maximum Weighted Independent Set (MWIS) problem. In this thesis, a new algorithm to the MWIS problem is introduced. The first part of the algorithm identifies vertices that belong to the MWIS using a $O(N)$ complexity algorithm, where N is the number of vertices of the graph. And the second part is a new greedy heuristic algorithm that is applied to the rest of the graph to find the approximate MWIS. It computational complexity is $O(E^C N^2 Dlog(N^2 D))$, where E is the maximum number of second degree neighbors of a vertex, C is the size of the allocated clusters, and D is the maximum degree of the graph. In the simulations ran, its performance was 89-97% for network sizes of 80 to 200 on bipartite graphs.

3 Introduction

This thesis seeks to find a computationally efficient solution to the problem of centralized dynamic spectrum allocation with the objective of maximizing social efficiency, subject to the constraint of interference. The ways of approaching this problem are diverse. In this thesis, we look at it as a maximum weighted independent set (MWIS) problem, where each vertex represents a user (transceiver), adjacencies between vertices on the graph represent interference, and vertex weights represent the bid values of the respective users. We seek to allocate to users such that the sum of weights of allocated vertices is maximized, and no two adjacent vertices are allocated. We then use VCG pricing to ensure incentive compatibility, where each bid value corresponds to the utility of the user.

The algorithm in this thesis for the maximum independent set problem consists of two parts. The first part identifies vertices that belong to the MWIS. The principle is that if a vertex's weight exceeds the sum of the weight of its neighbors, it belongs to the MWIS. This can be extended to a cluster of non-adjacent vertices. Each cluster consists of non-adjacent vertices, where each newly added vertex is separated from at least one vertex in the existing cluster by only one vertex. If the sum of the weights of the vertices exceeds the sum of the weights of all their neighbors (that do not belong to the cluster), they belong to the MWIS.

The second part of the algorithm allocates to the rest of the graph by such clusters, in descending order of (sum of bid values of clusters)/(sum of bid values of neighbors of clusters). We found that the performance of this algorithm surpasses that of allocating to single vertices, and does not increase computational complexity substantially, and hence is quick to implement. We have also found an efficient way of generating clusters, which further reduces computation time.

4 Background - Spectrum Market

Radio spectrum is utilized by television and radio broadcasts, cellular networks, wireless local area networks (WLAN), bluetooth networks, communications satellites, aviation, and radars, amongst others. It makes up the segment of the electromagnetic spectrum that is between 8.3 kHz to 3000 GHz in frequency.

The majority of radio spectrum is being allocated statically, where specific frequency bands are being reserved for specific services, and companies are allocated licenses for long-term usage of a particular bandwidth of a particular region. For example, 30-300MHz is reserved partly for radio and TV broadcasters, and 300-3000MHz is reserved partly for mobile phone usage and Wireless LAN [3], and each major telecommunications company like Verizon or AT&T owns licenses to certain frequency bands for certain geographical locations within the 300-3000MHz bandwidth. Companies can file for application for licenses electronically via FCC's Universal Licensing System. Since 1994, the FCC has typically used auctions to allocate licenses to commercial services [4]. Prior to that, comparative hearings and lotteries were being utilized to select a single licensee for a license [5]. The auction systems have advanced over the years, where package bidding is now available since the first combinatorial spectrum auction in 2007.

Mobile data transmission is undergoing rapid growth and the trend will persist. According to CISCO [1], Global IP traffic has increased fivefold over the past five years, and will increase three-fold over the next five years. The IP traffic growth from 2014 to 2019 will increase at an overall compound annual growth rate (CAGR) of 23%. This is due to multiple factors, two of which are the rapid adoption and development of various types of wireless technologies, including those of networks and devices. Moreover, the FCC also plans to provide affordable broadband access to every American, as well as to provide broadband services to anchor institutions such as schools, hospitals, and government buildings to every American community [6].

Currently, most of the spectrum allocation is done statically. Since spectrum utilization is time and space dependent, this form of allocation resulted in a substantial amount of white space existing in the spectrum.

To improve the allocation of radio spectrum to meet growing demands, various measures have been taken by the FCC. Spectrum has been redistributed from low usage TV broadcasters to higher usage users via an incentive auction, which is a double auction held by the FCC in May 2014. Secondary markets have also been formed where license holders can lease their spectrum to higher usage users for a particular period of time. These leases typically last for medium term, from any period less than a year, to several years. The different leasors in secondary include Mobile Virtual Network Operators, such as mobile cellular companies that operate on networks not of their own, Machine-to-Machine users, referring to data communication between remote machines, an example of which includes SCADA infrastructure, which are smart environments on large systems managed by large corporations or public entities such as smart power grids, and smart fresh and wastewater management systems [19], and explicit sale and lease [20].

Another important measure to meet growing spectrum demand is a class of spectrum access methods called Dynamic Spectrum Access (DSA). DSA makes use of cognitive radio technology to dynamically match demand with unused spectrum, hence eliminating white spaces and maximizing utilization of the spectrum. DSA involves the following components [18]: spectrum sensing where unused and hence available spectrum is being detected, spectrum management, where the best available channel is being chosen for access, and spectrum mobility that manages seamless transition between channels, and finally spectrum allocation.

There are multiple categories of DSA mechanisms. The chosen mechanism is dependent on the network infrastructure, and, if the spectrum is licensed, the interference threshold of the primary user. If there is a network infrastructure in place, the method of access will be a centralized one, where a centralized entity decides on the allocation of spectrum. In the case where it is not feasible to invest in infrastructure, the access method will be distributed where each secondary user accesses the spectrum based on the protocol or incentives in place. With regards to the second factor, if there is to be no interference to the primary user, the overlay technique will be used, where the appearance of a primary user will trigger a spectrum handoff. If, on the other hand, interference to the primary user is only to be limited by a threshold, the underlay technique will be utilized, where secondary users are allowed to use the channel concurrently as long as interference is below a certain threshold. [18].

5 Spectrum Allocation Methods

Only centralized spectrum allocation mechanisms will be covered in this thesis. Centralized spectrum allocation methods use a central controller to allocate such as to optimize the objective function for the overall network, as compared to decentralized/distributed methods where each user gathers, exchanges, and processes the information about the wireless environment independently before making the decision autonomously to access the spectrum. Centralized methods can be classified mainly into optimization methods and auction methods, with overlaps between the two methods [16].

5.1 Optimization Methods

With regards to optimization methods, graph coloring is commonly used, when the factor of fairness is being considered. Here, fairness refers to the state where every user gets allocated a minimum amount of spectrum (every vertex must be colored, and hence it is a graph coloring problem). This problem is similar to the channel assignment problem for cellular networks.

In [9], the spectrum allocation problem is formulated into three different objective functions: the first is to maximize total throughput, the second, to maximize minimum throughput, and the third is to maximize fairness. It uses a progressive minimum neighbor first algorithm. The proposed color-sensitive group coloring model is being formulated to maximize the channel utility according to the different objectives, and hence differs from the normal graph coloring problem. Each objective function is being met by assigning different labels to each vertex. For the first objective, in order to maximize throughput, the label for each vertex is its maximum valuation of a channel amongst all of its unallocated channels. To maximize minimum throughput (which will result in equal number of channels for all), the label for each vertex is the negative of the sum of its valuation for all channels it obtained. And to maximize fairness, the label for each vertex is the ratio of its maximum valuation for an unallocated channel to the sum of the valuations of all the channels it obtained.

In [10], the problem was also formulated as a graph coloring, and a max-k-cut problem. Each user i had two values in their demand function, specifying the minimum ($dmin(i)$) and maximum demand ($dmax(i)$) they had. Two objective functions were optimized. The first was to maximize overall demand met subjected to the interference constraint. The second is to minimize overall interference while meeting minimum demand for all users. In the first optimization problem, it is first checked that minimum demand for each user can be met, before demand met is being maximized. In the second optimization problem, the authors defined a new problem of multi-color max-k-cut, where each vertex is assigned to multiple different partitions (equal to the vertex's demand), such that sum of weights of edges crossing the partitions is maximized. They used the heuristic of randomly picking $dmax(i)$ colors from the available K colors for each vertex i, and showed that it yielded good results. They also used a Tabu search to improve the solution from the heuristic algorithm.

In [8], a Markov chain is being used to model DSA in the overlay mode. Arrival rates of both licensed and unlicensed users are assumed to be independent Poisson distributions. In order to account for the rejection of an unlicensed user when a licensed user is using the spectrum, waiting states are added in the Markov model. An admission probability for unlicensed users, $P(a)$, is used to prioritize spectrum access for licensed users. The objective function becomes the maximization of overall utility of all users, constrained by the expected utility of unlicensed users to be greater than zero. The optimization method in [11] uses the Coordinated Access Band (CAB) and Statistically Multiplexed Access concept, which aggregates users with ideally opposite time-varying spectrum demand. In the case when they are not aggregated, 2N channels are required where N is the average number of channels between the two, but with aggregation, fewer than 2N channels are required. In [7], the spectrum allocation problem was formalized as an optimization problem that maximizes the sum of the signal-to-interference (SIR) ratio of every user, subject to each user receiving a signal greater than its SIR threshold, and subject to total transmission power being lower than the interference temperature limit. This becomes a convex optimization problem.

5.2 Auction Methods

Computationally efficient spectrum auctions have been researched upon in recent years. However, most of them have used allocation mechanisms that allocate to individual bids in a descending order, which decreases social efficiency substantially. This is because a high bid may have two or more almost equally high bid in the adjacent vertices, and in such a case, allocating to the two lower bids will maximize their overall summed utility. For example, given a linear network of three vertices, i, j, k, with weights $w_i = 8, w_j = 9$, and $w_k = 8$, such a greedy algorithm will allocate to vertex j, attaining a network utility of 9, compared to the optimal network utility of 16, by allocating to i and k. There are multiple factors that a spectrum auction takes into account. The first is the properties of spectrum as a good. Spectrum is a reusable/divisible good, where a single unit can be used by multiple users within a geographical location, subject to interference constraints. Spectrum is also a multi-dimensional good, in terms of frequency, space, and time. The second factor is that the nature of demand for spectrum users may have combinatorial valuation, for example, valuing channels in adjacent geographical areas, to channels that are far apart. The third factor is general auctions properties, the most pertinent being strategyproof (truthfulness). A spectrum auction needs to ensure truthfulness of the bidders using a pricing mechanism to incentivize bidders to bid as high as they can afford, by ensuring they cannot pay less, nor, in the case of a non-monotonic allocation, obtain a channel by bidding lower. The fourth is the objective of the final allocation. Spectrum auctions can differ in their goals - some may aim to maximize social efficiency, some may aim at fairness, where each user gets allocated a minimum number of channels, and some may aim at maximizing revenue. The fifth is the computational complexity, which is important if allocation of channels is to be updated frequently.

The VERITAS mechanism was introduced in [12]. VERITAS assumes that bidders value each channel equally, and since it allocates according to descending value of bids, the channels are allocated sequentially among users. Hence, within each neighborhood, all except for one bidder will be allocated either all the channels they demanded for, or none at all. It uses secondary pricing to achieve strategyproofness. However, it allocates in descending order of bids and hence does not achieve optimal social efficiency.

The mechanism in [13] is similar, except that instead of secondary pricing, it uses the VCG mechanism where each winning bidder pays the opportunity cost it imposes on the network. It also utilizes a virtual valuation that normalizes bid values, which was first devised by Myerson [24]. A similar mechanism that utilized a decreasing marginal utility demand function for additional channels received for bidders was found in [17] In [15], the allocation to maximize social efficiency is modeled as an integer programming problem. The pricing mechanism uses secondary pricing to make it more computationally efficient, as compared to VCG.

A non-greedy allocation mechanism was used in the spectrum auction in [14], where a single channel is being allocated in a network that is divided into small hexagonal regions, and a knapsack approximation algorithm is utilized for the allocation, which results to be the set of non-interfering vertices which has the highest valued bids in all of the seven sets.

In this thesis, an auction mechanism where users demand multiple channels and the goal seeks to maximize overall network utility (maximize social efficiency) is considered. The assumptions made are that channels are homogeneous, interference ranges are constant, and users have decreasing marginal utility for each additional channel received. The winner determination problem is an NP-hard combinatorial problem of the maximum weight independent set [21] and this paper implements a new greedy algorithm that outperforms current greedy algorithms for the same problem, and employs the VCG mechanism to ensure strategyproofness.

6 The Maximum Weighted Independent Set Problem

6.1 Formulation

Given a graph $G(V, E)$, where V is the set of vertices and E is the set of edges in the graph, a subset x is called an independent set if no two vertices share an edge $(x_i, x_j) \neq (1, 1), \forall (i, j) \in E$. A maximum weight independent set is the heaviest weighted independent set of the graph. The MWIS problem can be formulated in terms of vertices, edges or sets. The most common formulation is the integer programming problem as follows:

$$\text{maximize} \quad \sum_{i=1}^{N} w_i x_i$$

$$\text{subject to} \quad x_i + x_j \leq 1, \quad \forall i, j \in E$$

$$x_i \in \{0, 1\}, \quad i = 1, 2, ..., N$$

A linear relaxation of the integer programming problem would relax the second condition into:

$$x_i \geq 0, \quad i = 1, 2, ..., N$$

The solution to the linear relaxation will yield $x_i = 0, .5, \text{or } 1 \quad i = 1, 2, ..., N$ [59]. The MWIS contains the vertex i for which $x_i = 1, \forall i$. A dual to the problem is as follows [59]:

$$\text{minimize} \quad \sum_{(i,j)\in E} \lambda_{ij}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{N}(i)} \lambda_{ij} \geq w_i, \quad \forall i \in \mathcal{V}$$

$$\lambda_{ij} \geq 0, \quad \forall i, j \in E$$

Shor(1990) [60] gave a quadratically constrained formulation for the MWIS, with the same objective function as the integer program:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} w_i x_i \\ \text{subject to} \quad & x_i x_j = 0 \quad \forall i,j \in E \\ & x_i^2 - x_i = 0, \quad i = 1, ..., N \end{aligned}$$

6.2 Related NP Hard Problems

6.2.1 Maximum Independent Set Problem

The MWIS problem can be approached from various angles. One might first look at the closely related problem of finding the maximum independent set (MIS) of a graph, which is a specific case of the MWIS problem where the weights all equal. They have the same constraints of which no two allocated vertices can be adjacent. In the integer programming formulation, they share the same constraints and objective function. Almost all methods for the MIS can be extendable to MWIS. The count of vertices, whether of a vertex's neighbors (the degree), or in an independent set, used as a factor in an algorithm to find the MIS can simply be replaced by sum of weights of a vertex's neighbors, or sum of weight of an independent set. Hence, these two problems also share equivalent lower bounds for certain algorithms.

6.2.2 Other Equivalent NP Hard Problems

The MWIS problem is equivalent to several NP hard problems. Hence, the solutions for these different problems are sometimes used interchangeably. The maximum weighted clique (MWC) of a graph, G, is equivalent to the MWIS of its complement graph, $\bar{G}$. The minimum vertex cover (MVC) and the maximum independent set (MIS) of a graph are disjoint and their union make up the entire graph. The same applies to the weighted version. Also, another name for the MWIS problem is the weighted vertex packing problem.

6.2.3 Other Related NP Hard Problems

There are other NP hard problems that are related to the MIS problem. They do not yield an equivalent solution to the MIS, but provide good bounds. They are, the graph coloring problem, clique cover problem, and the maximum matching problem. The minimum number of colors required to color a graph such that no two adjacent vertices have the same color, is the chromatic number of the graph. In a perfect graph, where there are no odd holes, the chromatic number is the same as the size of the maximum clique. In a non-perfect graph, the chromatic number is larger than the size of the maximum clique (consider a cycle of 5 vertices, where the maximum clique is of size 2 but the chromatic number is 3). The relationship between the chromatic number, $\chi(G)$ and the size of the MIS, $\alpha(G)$ of a graph $G(V, E)$, is $\chi(G) \geq \frac{|V|}{\alpha(G)}$, since the number of vertices in any color class defined by a proper coloring of G cannot exceed $\alpha(G)$ [36].

The minimum clique cover (MCC), or the clique cover number is the minimum number of cliques required to cover the graph. The clique cover number of a graph, G is equals to the chromatic number of its complement graph, $\bar{G}$, which is also equals to the smallest number of independent sets of $\bar{G}$. Hence, the minimum clique cover of a graph is an upper bound of the size of a maximum independent set. The size of the MIS cannot exceed the MCC because, for example, given a graph of n vertices, which can be covered by 2 cliques, or by 3 cliques, both of which are maximal, in the former case, only a maximum of 2 vertices can be allocated if we want them to be non-adjacent. In the latter case, this property remains unchanged, even though now there are 3 maximal cliques - there is no way of allocating to 3 vertices that are non-adjacent. In addition, the size of the MIS is not equals to the MCC in certain cases, for example in the case of an odd cycle. The maximum matching (MM) of a graph is the maximum number of edges to select without two edges sharing the same vertex. In a bipartite graph, the MM is equals to the minimum vertex cover, and the matching is perfect, where all vertices are adjacent to a selected edge. Hence, the MVC is an upper bound for the MM in general graphs.

6.3 Graph Types with Polynomial-time Exact Solutions

From a different angle of approaching the MWIS problem, there are specific graph types where it takes polynomial time to find the MWIS of a graph. The polynomial-time algorithms range from ellipsoid methods, to dynamic programming, to combinatorial algorithms on the dual of the linear programming relaxation of the original problem that are based on basic arithmetic operations and comparisons of rational numbers (t-perfect graphs [40]). Such graphs are: circle graphs (Gavril '73) [44], circular-arc graphs (Gavril '74) [43], claw-free graphs (Minty '80) [53], complements that contain no odd cycles (Hsu, Ikura, Nemhauser '81) [47], perfect graphs (Grotschel, Lovasz, Schrijver '84) where ellipsoid methods are used, claw-free or co-claw-free graphs (Brandstandt, Mahfud '02) [33], distance claw-free graphs (W.R. Pulleybank, F.B. Shepherd) [57] where dynamic programming is used, (p,q)-colorable graphs (Alekseev, Lozin '03) [26], fork-free graphs(Lozin, Milanic '08) [63], P6, co-banner - free graphs (Mosca '12) [54], hole and dart-free graphs (Basaravaju, Chandra '12) [31], P5-free graphs (Lozin, Mosca '08 , Karthick '13) [52] [49], and P4-free graphs (D.G. Corneil, Y. Perl, L.K. Stewart '85) [39]. There is also a linear-time algorithm for (P5,diamond)-free graphs (Brandstandt '02), [34].

6.4 Methods for Exact Solutions

In order to find an exact solution to the MWIS problem, apart from the brute force method, which enumerates all maximal independent sets of the graph and finds the highest weighted one, the most common and slightly more efficient method used is the branch and bound method. This is used to find either the MWIS of the graph, G, or the maximum weight clique on the complement graph $\bar{G}$. Other methods that yield exact solutions for NP hard problems include dynamic programming, preprocessing of data, and local search. For a comprehensive survey of exact methods for NP hard problems, one can refer to Woeginger (2003) [61].

Branch and bound is a search method that, instead of going down and completing every path of possible solutions, is able to prune a path midway by using a condition to determine that the rest of the path will not suffice to make the current set the solution (i.e. the bound fails to meet the condition). In our problem of the MWIS, the bounding function, is applied on the vertex that is last added to the independent set.

Branch and bound uses a search tree which is dynamically generated, where each node of the tree represents a subproblem. Hence, aside from the bounding function, which is the most crucial component, one must also decide how the original problem is to be broken up into subproblems, and the order of solving these subproblems [37]. These three decisions are interrelated and can vary for the same problem.

Branch and bound is more commonly applied to the MWC problem. To find a solution to the MWIS, we apply the method to find the maximum weighted clique to the complement graph, $\bar{G}$. In terms of the MWC problem, the root node is usually either the entire graph, or a single vertex. The search is a depth-first one. It builds cliques beginning with a single vertex, and adds a vertex in each iteration. The bounding function is usually the total sum of weights of the adjacent vertices of the last-added vertex, or the maximum weighted clique the last-added vertex is a member of, if it has previously been computed. The pruning condition is the weight of the maximum clique at that point in time. Then, instead of generating the entire maximal clique, it stops when the sum of the weight of the currently generated clique, and the weight of all the neighbors (or the maximum weighted clique) of the last-added vertex, does not exceed the weight of the current maximum weighted clique.

Even though the two problems of MWIS and MWC are equivalent, using the MWC to solve a MWIS problem may may not be feasible since the average degree of $\bar{G}$ is equals to $N - G$. Hence, if the average degree of G is < 10, $\bar{G}$ will have an average degree of more than $N - 10$. Finding a maximum clique of a graph with such a high average degree is significantly computationally heavier, even with branch and bound methods.

Branch and Bound (Carraghan/Pardalos '90; Ostergard '02) [38] [55] The best bounds for a maximum clique is obtained from graph coloring. However, there are also bounds obtained from clique formation, and these 2 seminal papers are the most cited for this method.

Both algorithms preorder the vertices according to a certain rule. It could be the number of degrees, or the order of coloring of the vertices according to a certain heuristic. In each iteration, both algorithms first add or remove a vertex, then finds the maximum clique that contains that particular vertex. In Carraghan/Pardalos's algorithm (CPA), the search begins with the entire graph, and with each iteration, removes a vertex, whereas in Ostergard's algorithm (OA), the search begins with a single vertex, of the largest index, N, and with each iteration, adds a vertex one index smaller, with the last vertex added being 1. For OA, in each iteration, the maximum clique is found with regards to the vertex that was added and the neighboring vertices that have a higher index than it. For CPA, the procedure is exactly the same, except in the reverse order, where it finds the maximum clique with regards to vertex 1 and its neighboring vertices that have a higher index than it, in the first iteration. What differs between the 2 algorithms is the bounding function used. OA starts from vertex N, and uses the size of the clique formed by the newly added vertex from each iteration, in the bounding function for subsequent iterations. In CPA, the bound is simply a function of the degree of the newly removed vertex. Hence, OA uses a tighter bound.

In order to extend either algorithm to the weighted case, in the bounding function, the degree of a vertex is replaced with the sum of weights of a vertex's neighbors in CPA, and the size of a clique is replaced with the weight of a clique in OA.

Branch and Bound (Kumlander '05) [50] Kumlander's algorithm for finding the MWC is similar to Carraghan and Pardalos's method. It also starts off with the entire graph, and with each iteration, removes a vertex. In each iteration, it also finds a maximum clique with regards to the vertex with the highest index. However, for the bounding function, instead of using degree of the vertex, it uses the number of color classes of the subgraph, which is the neighborhood of the vertex i. The exact bounding function is: prune if $d - 1 + D < C$, where D is the degree, computed by the sum of the maximum weight of every color class in that subgraph. The degree will not be recomputed at each depth level, but will be adjusted according to this rule: if the next vertex to be expanded on the current depth level is of the same color as the current vertex, the degree will not be increased. Otherwise, the weight of the current vertex should be deducted from the degree. Kumlander employs a second pruning strategy obtained from Ostergard's algorithm as illustrated previously, using $c(i)$.

Branch and Bound (Babel '90) [27] Babel extends a concept by Brelaz, and uses a measure called the Generalized Clique Degree (GKD) of a vertex, or subset, which is the sum of cliques that are formed by its non-neighbors. It measures the least number of cliques that must be added to its non-neighbors to form a clique cover. The algorithm keeps track of a list of cliques, and with each iteration, chooses an uncovered vertex with the highest GKD and adds it to a clique that it belongs to. If no such clique exists, a new one is formed. Since the vertices are selected in order of GKD, the members of some maximal independent set are the first vertices to be selected.

This algorithm uses weighted clique covers as upper bounds. The rest of the procedure is like above. The algorithm starts by pre-ordering the vertices in ascending order of the summed weight of a vertex's neighbors.

Branch and Bound (Balas and Yu '89) [28] Balas and Yu starts off with finding a maximal induced triangulated subgraph (MTIS) of graph $G, G(T)$, by checking to see if G admits a perfect ordering. This is done by first locating a simplicial vertex. It then finds a maximum clique ($C*$) of $G(T)$, and then a minimum coloring for $G(T)$ (which is the same cardinality, k, as the maximum clique). It then extends $G(T)$ by adding vertices to it without increasing the number of color classes, to achieve the maximal k-chromatic induced subgraph, $G(W)$. Branch and bound is conducted for the rest of the graph, $V \backslash W$. Since any clique that is larger than $C*$ must contain at least a vertex from $V \backslash W$, the branch and bound is conducted on neighborhoods of vertices from $V \backslash W$.

Branch and Bound (Balas and Xue '90) [29] The paper uses a method that is in a similar vein to that used by Balas and Yu. It uses weighted clique covers as an upper bound for the weight of the maximum weight independent set. It starts off with a benchmark weight of a randomly chosen independent set, F. Then, an induced subgraph,S, of G, of which the maximum weight independent set is smaller than that of F, is chosen. Each vertex, v, in $G - S$ is indexed, and for $\bar{N}(v(i)) \cap \{v(j) : j<i\}$, we find the maximum weight independent set, $S(i)$. The MWIS of graph G is then either F, or $v(i) \cup S(i)$.

The first step of finding an induced subgraph of which the maximum weight independent set is smaller than that of F is done using clique covers. Each vertex in the independent set F is a member of a distinct clique, and for each clique, we add a vertex to a clique if all vertices in the clique are neighbors of the vertex. We add as many vertices as possible this way, and the resultant induced subgraph containing all these vertices will fulfill the criterion in step 1.

6.5 Methods for Approximate Solutions

6.5.1 Greedy Heuristics

Two simple greedy heuristics have been used for the MWIS. The first is to allocate vertices in descending order of the ratio of the vertex weight to the sum of weights of their adjacent vertices, and to remove all adjacent vertices of the allocated vertex. The other is to remove vertices and their edges in ascending order of that value, until no edges remain - the remaining vertices form the independent set. In the Caro-Wei theorem, it has been proven that both the first and second algorithms achieves a performance guarantee of an independent set with weight $\sum_{v \in V(G)} \frac{W(v)}{d(v)+1}$, where $d(v)$ is the count of adjacent vertices of a vertex. [58]. Our algorithm, by allocating to clusters in a greedy manner, is able to achieve a bound of at least $\sum_{v \in V(G)} \frac{W(v)}{d(v)+\frac{1}{c}}$ by applying the same principle, since there will be a least $c-1$ shared adjacent vertices in a cluster of size c - hence with every allocation of a cluster, C, of size c, $(c \times (d_i + 1) - 1, i \in C)$ vertices will be removed from the graph.

6.5.2 Metaheuristics

Metaheuristics allow a local search algorithm to escape local optima, in search of a better solution or the global optimum.

Genetic Algorithm (Hifi '97) [46] Hifi came up with a genetic algorithm to address the maximum weight independent set problem. The fitness function is the sum of the sum of weights and a penalty for constraint violation of a given solution to the MWIS problem. A solution is an independent subset, represented by a binary vector, where each vertex, i is 1, if it belongs to the solution, and 0 otherwise. In order to generate feasible solutions where the interference constraint is not violated, each solutions is generated by picking a random vertex, and removing the adjacent vertices from the set, and repeating the process. The probability of a solution being selected as a parent is the difference of its fitness function and the minimum fitness function, divided by the sum of this difference for all solutions. Two parents will then produce two children using the following crossover operator:

Given a solution to a graph, G, of N vertices, of parent $i = P_i = [P_{i,1}, P_{i,2}, ...P_{i,N}]$, and that of a child, $i = C_i = [C_{i,1}, C_{i,2}, ...C_{i,N}]$, then given 2 parents, 1, and 2, and 2 children, 1, and 2,

1. If $P_{1,i} = P_{2,i}$, then $C_{1,i} = C_{2,i} = P_{1,i}$

2. If $P_{1,i} \neq P_{2,i}$, then randomly attribute to k and $k'(k \neq k')$ a value in $\{1, 2\}$, and

    a. set $C_{1,i} = P_{k,i}$ with probability $p_k = \frac{f_k^f}{f_k^f + f_{k'}^f}$ and $C_{1,i} = P_{k',i}$ with probability $1 - p_k$

    b. set $C_{2,i} = P_{k',i}$ with probability $p'_k = \frac{f_{k'}^f}{f_k^f + f_{k'}^f}$ and $C_{1,i} = P_{k,i}$ with probability $1 - p_{k'}$.

The mutation operator follows, by inverting M randomly chosen variables in the binary vector of each child, where M is determined experimentally. This procedure removes violations in constraints in solutions, and adds feasible vertices to a solution. It then computes the new fitness value. Finally, less fit parents will be replaced by fitter children in the population, using the incremental replacement method.

Simulated Annealing In Feo, Resende and Smith (1989)'s paper [42], a simulated annealing approach to solving the maximum independent set is described. The graph is partitioned into two subsets, K and $\bar{K}$. The objective is to minimize the function $f(K)$ = -|K| + lambda*$|E_k|$, where E_k is the set of edges in the induced subgraph K. A neighbor solution, $K'(v)$ is generated by choosing a random vertex and adding it to K if it is in $\bar{K}$, and removing it from K if it is already in K. The neighbor solution is kept if it does not worsen the objective function and if it does, it is kept with a probability of $e^{\frac{-d}{T_k}}.d = f(K'(v)) - f(K)$ and T is the temperature, which is decreased in accordance with a cooling schedule.

6.5.3 Local Search Heuristics

Local search methods work by perturbing the initial solution to seek a better solution that satisfies the constraints.

Greedy Randomized Adaptive Search Process (Feo, Resende and Smith '89) [42]

Here, the GRASP algorithm was used to find a maximum independent set. It is a three-step procedure that first involves an adaptive ordering of the vertices that are eligible for selection in the next step. The vertices are first ordered in ascending order of degrees, then, the first x number of vertices will form pairs, and for each pair, the number of non-neighbors from the entire graph of either of the vertices is computed, and the pairs are ordered in descending order of this number. The orderings are updated after each selection. In the next step, there is a probabilistic selection from the top x elements of the sorted list, followed by a local search to perturb the current solution to see if there is a better one – the method attempts to remove k vertices from the current solution and search for k+1 vertices as replacement. The algorithm can be run in parallel by splitting the graph into subsets of vertices.

TABU Search (Wu, Hao and Glover '12) [62] In this paper, a Tabu search for the maximum weighted clique (MWC) is described, which outperforms all previous methods for MWC. The algorithm starts with a randomly selected vertex, followed by the formation of a maximal clique the vertex belongs in (by the selection of vertices that are adjacent to all previously selected vertices). There are then three types of moves that can be made to perturb the maximal clique – a vertex can be added to, dropped, or swapped from the clique, as long as the clique remains as a clique. A move is selected based on the vertex weight value it adds to the clique. Another caveat is that a vertex that has been dropped from the clique cannot be added in the next x iterations. Finally, the algorithm uses a multistart strategy where the search is restarted in a new location when the current search is deemed to be stuck in a local optima, which is judged based on its inability to be improved after a pre-determined number (depth of search) of consecutive iterations.

6.5.4 Scale Reduction Methods for Approximate Solution

Scale reduction methods seek to identify a subset of the MWIS of a graph, or a subset that does not belong to the MWIS, before applying search methods to find the rest of the graph.

Identifying vertices in MWIS: Critical Sets A critical weighted set, X, is one that has the maximum value of $w(X)$–$w(N(X))$, where $N(X)$ is the set of vertices that is adjacent to at least one vertex in X. This set can be found in polynomial time, and the isolated vertices in this set belong to the MWIS. It is shown in Zhang (1990)'s paper [64] that the minimum vertex set is an independent set. In Butenko's (2003) paper [35], he proved that a critical weighted independent set is a subset of the MWIS.

In Zhang's (1990) paper, he proved that the critical weighted independent set of a graph can be found in polynomial time, by conducting a bijection mapping of the graph to form a bipartite version of the graph, G, also known as the bi-double graph of $G, (G', E')$. A bi-double graph is simply made of V and a duplicate set of V, V', and there is an edge between $V(u)$ and $V'(v)$ if

$(u, v) \in E$, and $(V(u), V(v)) \notin E', (V'(u), V'(v)) \notin E', \forall u, v \in V$. In Zhang's paper, he solved it as a linear programming problem in polynomial time. In Ageev (1994)'s paper [25], it was mentioned that finding the critical weighted independent set can be solved as a maximum flow or minimum cut problem (Balinski '70) [30], which can be done in polynomial time. The edges from the source to each $V(v)$, and from V'(v) to the sink, $\forall v \in V$,is the weight of each vertex, v, and each edge in E', $(V(u), V'(v)) \in E'$, has a weight of infinity. The maximum flow from the source to the sink is the minimum cut of the graph. It is also the minimum weighted vertex cover, and its complement is the maximum weighted independent set.

The limitation of this method is that sometimes the critical set may be an empty one. For example, in a 3-cycle graph, K_3, the minimum cut yields the full set if none of the vertices have a weight that exceeds the sum of weights of the adjacent vertices.

Elimination: Removing Subgraphs (Feige '04) [41] Feige came up with an algorithm that improved the approximation results of a maximum clique from Halldorsson's approximation of $O(n/(log(n))^2)$ to $O(n(loglog(n))^2/(log(n))^3)$, which is as good as the approximation for chromatic number.

This algorithm is about eliminating parts of the graph that do not belong to the maximum clique. Let $G(V, E)$ be a graph with n vertices, which contains a clique of size $\frac{n}{k}$. It breaks the graph up into smaller disjoint subgraphs, $S_1, S_2, ...$, and in each subgraph, it seeks to identify poor subgraphs. A subgraph, S, is defined to be poor if it does not contain a clique of size greater than $\frac{|S|}{2k}$. Each round of the iteration searches for one of the following two outputs: either a clique of size $tlog_{3k}\frac{|V|}{6kt}$, or a poor subgraph. If it's the first, the algorithm terminates, and if it's the second, it will be removed from the graph. The search is conducted by breaking the graph into smaller subsets of size t. For each set, if it is a clique, and if the subset outside the clique which is connected to every vertex in the clique is of size greater than $\frac{1}{2k-t}$ of the subset, then the clique belongs to the maximum clique and the set of neighbors is used as the next input. The algorithm terminates when all the subsets have been searched.

7 Problem Definition

This thesis presents a centralized allocation mechanism involving a sealed-bid auction that is computationally efficient and allocates based on a heuristic algorithm that achieves an approximately socially optimal outcome. It is priced using the Vickrey-Clarke-Groves (VCG) mechanism. It will be shown that this mechanism achieves an ϵ degree of strategyproofness. This will be defined in section 5.

The problem is defined as follows:

Let $\mathcal{V} = \{1, \ldots, N\}$ be the index set of vertices of the network, where each vertex represents a user (transmitter and receiver pair). We define the *interference graph* $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ and arc or edge $(i, j) \in \mathcal{E}$ exists between vertices $i, j \in \mathcal{V}$ if joint channel exploitation by users i and j leads to unacceptable levels of interference. The neighborhood of vertex i, $N(i)$, is the set of vertices that interfere with i, i.e., $N(i) = \{j \in \mathcal{V} | \, (i, j) \in \mathcal{E}\}$. There is a set of channels $\mathcal{M} = \{1, \ldots, M\}$ available for use. A channel allocation is a map $x_{i,m} : \mathcal{V} \times \mathcal{M} \mapsto \{0, 1\}$, i.e. if $x_{i,m} = 1$, vertex is allowed use of channel $m \in \mathcal{M}$.

Let $v_{i,m}(x)$ be defined as follows

$$v_{i,m}(x) = \begin{cases} v_{i,m} & x_{i,m} = 1 \text{ and } \sum\limits_{j \in N(i)} x_{j,m} = 0 \\ \\ 0 & \text{otherwise} \end{cases}$$

where $v_{i,m} > 0$ represents the utility user i gets from using channel m *without* interference. The aggregate utility user i obtains from m channels can be expressed as

$$U_i(x) = \sum_{m \in \mathcal{M}} v_{i,m}(x^*_m)$$

During bidding, each user, i, will report its bid for each additional channel it receives, $\mathbf{b}_i = [b_{i,1}, ..., b_{i,M}]$. We say that a bidder, i, is truthful if its bids equal to its valuations, i.e., $\mathbf{b}_i = \mathbf{v}_i$.

The optimal allocation for a single channel, m, is

$$x_m^* = \underset{x}{\operatorname{argmax}} \sum_{i \in \mathcal{V}} b_{i,m}(x) \tag{1}$$

And the optimal allocation for all channels is represented by

$$\mathbf{x}^* = [x_1, x_2, ..., x_{\mathcal{M}}] \tag{2}$$

The incentive for bidders to be truthful is provided by the VCG payment rule, which will be elaborated on in section 5.

8 Methodology

8.1 The Allocation Mechanism

In the first part of our algorithm, we allocate to vertices which have weights that exceed the sum of the weights of their neighbors. We prove here that these vertices belong to the MWIS:
Let w_i be such that $w_i \geq \sum_{j \in \mathcal{N}(i)} w_j$. Given an arbitrary initial allocation, assume that i is not allocated, and some of its neighbors are. By allocating to i and deallocating the neighbors which have been initially allocated, we can be certain that overall network utility, $\sum_{i=1}^{N} w_i$, will not decrease, since what was deallocated cannot exceed $\sum_{j \in \mathcal{N}(i)} w_j$ and $w_i \geq \sum_{j \in \mathcal{N}(i)} w_j$. The same principle applies to a cluster of vertices, where there is a maximum of one vertex in between a member of the cluster and at least one other member.

A greedy algorithm that allocates to multiple vertices/nodes per iteration, the Multi-Node Greedy Allocation Alogirthm (MNGA) that attains a good approximation of the socially optimal solution is presented in the following. The notion of *dth*-degree neighborhood and clusters is first introduced. The *dth*-degree neighborhood of vertex i, $N_d(i)$, is the set of vertices whose shortest path from i involves d edges. A cluster of vertices, i, of size c, $Q_i(c) = \{q_1, ..., q_c : q_i \in V | q_i \in N_2(q_{i1}), q_j \notin N(q_{i \neq j})\}$, are vertices that are each even-degree neighbors of each and every vertex in the cluster, and the cluster value, $S(c) \in \mathbb{R}$, is the sum of the maximum bid value of each of the c vertices, i.e., $\sum_{i \in \mathcal{Q}(c)} max(\mathbf{b}_i)$.
Channels that are available for bidding for each vertex i is represented by the set $\mathcal{C}_i = \{z_{i,m} : i \in \mathcal{V}, m \in \mathcal{M}\}$, where $z_{i,m} = 1$ if the channel m is available to i.

Now consider the following allocation algorithm for a single channel (read all $\mathbf{b}_i$ or max($\mathbf{b}_i$) as b_i). Form size-c clusters, starting with c = 2. Sort the clusters in descending order of the sum ratio (ratio of the summed weights of a cluster to the summed weights of the neighbors of the cluster), or count ratio (ratio of the number of vertices in the cluster to the total number of neighbors of the cluster) . Increase c by 1 until $c = 5$. Use cluster size with best performance. This procedure is repeated for each of the m channels.

Algorithm 1- Size-2 cluster Value Computation

Input: *G(V,E,b)*

Output: Sum or count ratio of every pair of "alternate" nodes, i, j, $i \in \mathcal{N}_2(j)$

For $i \in V$, **for** $j \in V$,

If $(i, j) \in E$, **then** $j \in \mathcal{N}(i)$ **and** $i \in \mathcal{N}(j)$

For $i \in V$, **for** $j \in \mathcal{N}(i)$, **for** $k \in \mathcal{N}(j)$

If $k \notin \mathcal{N}(i)$, **then** $k \in \mathcal{N}_2(i)$

For $i \in V$, **for** $k \in N_2(i)$,

$S_{i,k}(2) = \frac{max(b_k)+max(b_i)}{\sum\limits_{j \in \mathcal{N}(i) \cup \mathcal{N}(k)} max(b_j)}$

End

Algorithm 2- Size-c cluster Value Computation (c > 2)

Input: *G(V,E,b)*, Size-*(c-1)* Clusters

Output: Sum or count ratio of every size c cluster of nodes, C, where $C = \{i : i \in \mathcal{N}_2(j), i \notin \mathcal{N}(j) | j \in C\}$

For each size-*(c - 1)* cluster, where the last added vertex was j, **for** each vertex $k \in N_2(j)$,

If $k \notin N(Q(c-1)_{i,j})$, **then** $S(c)_{i,k} = max(b_k) + S(c-1)_{i,j}$

End

Algorithm 3- Multiple Channel Allocation

Input: *G(V,E,b)*, Size-*(c-1)* Clusters Ranked According to Sum of Weights of the *(c-1)* Nodes

Output: Greedy Allocation for m Channels via Size-c Clusters

For each channel m,

Compute $S(c)$

$Q'(c)$ = Sort $Q(c)$ in descending order of values of $S(c)$

For all vertices i in Top($Q'(c)$),

If $z_{i,m} = 1$, **then** $x_{i,m} = 1$ **and** $z_{i,m} = 0$ **and** $b_i = b_i \backslash max(b_i)$

For all vertices $j \in N(i)$,

$z_{j,m} = 0$

End

The computational complexity for generating the pairs is $O(N^2 Dlog(N^2 D))$, where D is the maximum degree of the graph, and the computational complexity for generating clusters of size C is $O(E^C N^2 Dlog(N^2 D))$, where E is the maximum number of second degree neighbors of a vertex.

In deciding the size of the cluster to be used for the allocation, since no single size consistently outperformed the rest, all cluster sizes should be generated and the best performing cluster size be used.

8.2 The Pricing Mechanism

As for pricing, the Vickrey-Clarke-Grove auction mechanism is used to maximize the degree of strategy-proofness. For each channel m, each allocated vertex i, will be charged

$$\max_{x} \sum_{j \neq i \in \mathcal{V}} b_{j,m}(x) - \sum_{j \neq i \in \mathcal{V}} b_{j,m}(x^*) \tag{1}$$

And the total payment for vertex i will hence be

$$\sum_{m \in \mathcal{M}} (\max_{x} \sum_{j \neq i \in \mathcal{V}} b_{j,m}(x) - \sum_{j \neq i \in \mathcal{V}} b_{j,m}(x^*)) \tag{2}$$

It will first be shown that the VCG auction mechanism is strategy-proof for the optimal allocation for one channel.

To be strategy-proof, it must deter underpayment caused by untruthful bidding in all 3 possible scenarios: If v_i is such that vertex i will be unallocated if it were truthful, then if $b_i < v_i$, vertex i must not be allocated the channel, and if $b_i > v_i$ such that he now gets allocated at the expense of another vertex, his new payment must not be $\leq v_i$. Finally, if v_i is such that vertex i will be allocated if it were truthful, then by reporting any $b_i \neq v_i$, the charge for his allocation must not decrease.

The proof for the first two scenarios is as such:
Let there be a vertex k with bid $b_{k,m} = v_{k,m}$, that does not belong to the optimal allocation:

$$b_{k,m} + \max_{x} \sum_{j \neq k \in \mathcal{V} \setminus \mathcal{N}(k)} b_{j,m}(x) < \sum_{i \in \mathcal{V}} b_{j,m}(x^*) \tag{3}$$

For all $b'_{k,m} < v_{k,m}$,

$$b'_{k,m} + \max_x \sum_{j \neq k \in \mathcal{V} \setminus \mathcal{N}(k)} b_{j,m}(x) < \sum_{i \in \mathcal{V}} b_{j,m}(x^*) \tag{4}$$

And hence vertex k will not be allocated if it bids lower. For the case where $b'_{k,m} > v_{k,m}$ such that vertex k now belongs to the optimal allocation, x'*:

$$b'_{k,m} + \max_x \sum_{j \neq k \in \mathcal{V} \setminus \mathcal{N}(k)} b_{j,m}(x) = \sum_{i \in \mathcal{V}} b_{j,m}(x'^*)$$

$$\sum_{i \in \mathcal{V}} b_{j,m}(x'^*) > \sum_{i \in \mathcal{V}} b_{j,m}(x^*)$$

Since

$$\max_x \sum_{j \neq k \in \mathcal{V}} b_{j,m}(x) = \sum_{i \in \mathcal{V}} b_{j,m}(x^*) \tag{5}$$

The payment charged to vertex k will be

$$\sum_{i \in \mathcal{V}} b_{j,m}(x^*) - (\sum_{i \in \mathcal{V}} b_{j,m}(x'^*) - b'_{k,m}) \tag{6}$$

Since in order to change an unallocated bid into an allocated one,

$$b'_{k,m} - b_{k,m} \geq 2 \tag{7}$$

and

$$\sum_{i \in \mathcal{V}} b_{j,m}(x^*) - \sum_{i \in \mathcal{V}} b_{j,m}(x'^*) < -(b'_{k,m} - b_{k,m} - 1), \tag{8}$$

k must pay at least $b_{k,m} + 1$ to be allocated channel m.

The proof for the third scenario is as follows: The payment for an allocated vertex i is

$$\max_x \sum_{j \neq i \in \mathcal{V}} b_{j,m}(x) - \sum_{j \neq i \in \mathcal{V}} b_{j,m}(x^*) \tag{9}$$

Hence, it can be seen that the payment is independent of $b_{i,m}$, and by lowering its bid, it will not be able to pay less.

8.3 ϵ-Strategyproofenss

Since this allocation is an approximate one, it is only approximately strategyproof. The concept of ϵ-strategyproofness is from Kothari, Parkes, Suri [22]. A mechanism is ϵ-strategyproof if a bidder can gain at most ϵ through non-truthful bidding.

Since, as shown, the optimal allocation is strategyproof when the VCG pricing mechanism is applied, it can be shown that this mechanism attains an ϵ-degree of strategyproofness, where $\frac{1}{1+\epsilon}$ is the degree of approximation of the allocation. Let $\hat{V}$ denote the allocation of our approximating allocation mechanism, and V^* denote the optimal allocation. Then assuming the approximation fulfills: $\hat{V} < V^*$, a vertex i can try to improve his payoff by bidding $b_i \neq v_i$ such as to change the outcome from $\frac{V^*}{1+\epsilon}$ to V^*. Hence, vertex i's gain in utility is $\frac{\epsilon}{1+\epsilon}V^*$.

8.4 Extension to Multiple Channels

The allocation and pricing processes described above can be repeated for each channel without making any adjustments, since the channels are independent of one another in terms of interference. The properties of social optimality and truthfulness are thereby maintained. The additional factor of consideration for multiple channels is the demand of bidders for different channels. It is assumed that each bidder values an additional channel at an equal or lower value than the previous channel. Hence, the bids per bidder for different channels will be marginally decreasing in m. During the auction, a bidder's highest bid will be removed from his bids each time he attains a channel. The bids for each bidder will then be updated during each round of allocation for a new channel.

9 Results of Allocation Mechanism

Simulations were conducted using retangular-grid topology, bipartite graphs, and weighted DIMACS graphs using weights used in Pullan's paper. Because this algorithm makes use of the extra information provided by the weight of each vertex in the graph to find the optimal MWIS, the usual simulations run for MIS algorithms were not being used here. Simulations were first conducted on a rectangular-grid topology, with odd number of columns and rows. The bid values are uniformly distributed with a mean of 20 and variance of 1. In this way, the socially optimal allocation is likely to be that of alternate vertices beginning with the corner vertex, since this maximizes the number of vertices being allocated. It is shown that allocation by cluster sizes of at least 2 consistently outperforms that of single vertices, and allocating greedily using clusters achieves an approximation degree of 90 to 100 %, for network sizes of 49 to 361. The optimal cluster size (one that achieves the socially optimal allocation) has a positive correlation with the network size, up till a cluster size of 5.

Network size (no. of vertices)	Optimal cluster	Degree of approximation	Degree of approximation of single vertex	Degree of approximation of single vertex/count
49	c/s(3)*; c/s(4)	100%	70.9%	85.1%
81	c/s(4)	100%	74.4%	82.8%
121	c/s(5)	96%	74.9%	81.2%
169	c/s(5)	92%	76.0%	82.1%
361	s(6)**	90.7%	76.8%	80.4%

*c/s(i) stands for both count and sum ratios of cluster size i

*s(i) stands for sum ratio of cluster size i

Simulations were next conducted on planar networks by varying 4 variables:

- Network size, N
- Mean number of neighbors, Mn
- Variance of number of neighbors, Vn
- Variance of bid values, Vb

Other than testing for varying cluster sizes, we also included another benchmark method of allocating to alternate vertices to test the effectiveness of our method, overall. We ran 20 simulations each for $N = 100$ with varying Mn, Vn and Vb. We used Mn for 5%, 10% and 15% of N, Vn for 30%, 40% and 55% of Mn, and Vb of 15%, 40%, and 50% of the mean values of the bids. In general, there is no determining factor for best cluster size, but allocation by clusters outperforms allocation by single vertices 90% of the time. In addition, the ratio using count, instead of sum of neighbors, worked better for higher bid variances, and when variance of number of neighbors increases, the method of allocating to alternate vertices tend to outperform that of the greedy allocation method. Mean number of neighbors does not have a detectable impact on the above mentioned factors.

In order to test for the performance of the clusters with respect to the optimal solution for the MWIS problem, we ran simulations for 5 DIMACS graphs, and for the vertex weights, we used the values that Pullan [56] used for the DIMACS-W set, which is $i \times mod(200) + 1$ for vertex i.

C125	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
Sum Ratio	88%	92%	91%	94%	92%	**94%**
Count Ratio	90%	**97%**	94%	92%	94%	**97%**
Difference	84%	84%	89%	89%	78%	**89%**

brock200−4	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
Sum Ratio	81%	81%	83%	72%	72%	83%
Count Ratio	81%	80%	69%	87%	78%	87%
Difference	61%	71%	85%	51%	69%	85%

Mann−27	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
Sum Ratio	98%	99%	101%	99%	99%	100%
Count Ratio	100%	98%	99%	99%	99%	100%
Difference	99%	98%	99%	99%	88%	99%

Gen200−4	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
Sum Ratio	89%	86%	92%	80%	87%	92%
Count Ratio	80%	84%	87%	85%	83%	87%
Difference	80%	84%	87%	85%	83%	87%

Simulation was also done using graphs of random edge densities within specified ranges, and Gaussian distributed vertex weights. The graphs were bipartite, which would not affect the performance of this particular algorithm, since the factors affecting the performance of the algorithm are network size, and possibly variance of vertex weights, and the variance in density of edges of the graph, given the principles behind how the algorithm functions. These three factors can be adjusted in a bipartite graph. On the other hand, if this were an algorithm that involves enumeration, a bipartite graph would improve the implementation time of the algorithm since it decreases the number of possible combinations of allocated nodes in an independent set of the graph.

The optimal solution of the MWIS of the bipartite graphs were obtained by using both the max-flow and min-cut algorithms [25] from the matlab_bgl package, which runs in polynomial time. However, they yield different results most of the time. The average degree of the networks simulated is 3% of the network size, and the variance to mean ratio of the degree of the vertices is approximately 1.3% of the network size. Below are the results of 180 simulations for each network size(table):

% **Performance Compared With Optimal**

Performance of Sum-Ratio:

N	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
80	76%	82%	86%	88%	89%	**93%**
120	76%	84%	88%	83%	78%	**91%**
160	76%	80%	88%	88%	92%	**96%**
200	75%	80%	84%	84%	95%	**96%**

Performance of Count-Ratio:

N	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
80	80%	87%	88%	89%	91%	**93%**
120	70%	77%	82%	85%	97%	**97%**
160	78%	81%	85%	83%	88%	**91%**
200	73%	79%	86%	86%	88%	**89%**

Average Performance For Sum-Ratio

Var-mean Ratio	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
4%	75%	81%	84%	87%	85%	**92%**
8%	76%	82%	87%	89%	88%	**94%**
12%	76%	81%	85%	85%	96%	**96%**

Average Performance For Count-Ratio

Var-mean Ratio	Single vertex	Size-2 Cluster	Size-3 Cluster	Size-4 Cluster	Size-5 Cluster	Best-Of MNGA
4%	75%	81%	87%	84%	91%	**92%**
8%	75%	81%	86%	87%	91%	**93%**
12%	76%	81%	84%	87%	90%	**91%**

10 Conclusion

In this paper, a computationally efficient algorithm and strategyproof pricing mechanism for allocating radio spectrum, with the objective of maximizing social utility, subject to interference constraints, is presented. A two-part allocation algorithm for the Maximum Weighted Independent Set was introduced, which outperforms the single-vertex greedy heuristic 91% of the time, and achieves a performance of 89-97% of the optimal solution on average, as compared to single-vertex allocation's performance of 70-80% for network sizes of 80 to 200. It works well for networks with high variance to mean ratios in weights. The computation complexity of this algorithm is $O(E^C N^2 Dlog(N^2 D))$, where E is the maximum number of second degree neighbors of a vertex. This can be further reduced by applying bounds to the number of clusters being generated.

We believe that this heuristic, aside from being used to find a good approximate solution, can be highly useful in finding a good initial solution for local search and metaheuristic algorithms for MWIS, MIS, MWC, Maximum Clique, and graph coloring problems.

References

[1] CISCO. “The Zettabyte Era: Trends and Analysis White Paper” May 2015

[2] International Telecommunication Union Radio Regulations, Volume 1, Article 2; Edition of 2008

[3] National Telecommunications and Information Administrative Office of Spectrum Management. “Federal Spectrum Use Summary.” June 21, 2010

[4] http://www.fcc.gov/encyclopedia/accessing-spectrum

[5] http://wireless.fcc.gov/auctions

[6] Federal Communications Commission. “The National Broadband Plan.” March 17, 2010.

[7] Y. Xing, C.N. Mathur, M.A. Haleem, R. Chandramouli. “Dynamic spectrum access with QoS and interference temperature constraints.” Mobile Computing, IEEE Transactions on 6.4 (2007): 423-433.

[8] B. Wang, Z. Ji, K.J.R. Liu. “Primary-prioritized Markov approach for dynamic spectrum allocation.” Wireless Communications, IEEE Transactions on 8.4 (2009): 1854-1865.

[9] H. Zheng, and C. Peng. “Collaboration and fairness in opportunistic spectrum access.” Communications, 2005. ICC 2005. 2005 IEEE International Conference on. Vol. 5. IEEE, 2005.

[10] A.P. Subramanian, H. Gupta. “Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks.” New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on. IEEE, 2007.

[11] T. Kamakaris, M.M. Buddhikot, R. Iyer. “A case for coordinated dynamic spectrum access in cellular networks.” New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on. IEEE, 2005.

[12] X. Zhou, S. Gandhi, S. Suri, H. Zheng. "eBay in the sky: strategy-proof wireless spectrum auctions." Proceedings of the 14th ACM international conference on Mobile computing and networking. ACM, 2008.

[13] J. Jia, Q. Zhang, Q. Zhang, M. Liu. "Revenue generation for truthful spectrum auction in dynamic spectrum access." Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing. ACM, 2009.

[14] Al-Ayyoub, Mahmoud, H. Gupta. "Truthful spectrum auctions with approximate revenue." INFOCOM, 2011 Proceedings IEEE. IEEE, 2011.

[15] Y. Wu, B. Wang, K.J.R. Liu, T.C. Clancy. "A multi-winner cognitive spectrum auction framework with collusion- resistant mechanisms." New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on. IEEE, 2008.

[16] E. Hossain, D. Niyato, Z. Han. "Dynamic spectrum access and management in cognitive radio networks." Cambridge University Press, 2009.

[17] J. Barrera, A. Garcia, A, M. Hong. "Auction design for spectrum allocation un- der interference constraints." Global Communications Conference (GLOBECOM), 2013 IEEE. IEEE, 2013.

[18] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanti. "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey." Computer Networks 50.13 (2006): 2127-2159.

[19] Chapin, M. John, W.H. Lehr. "SCADA for the Rest of Us: Unlicensed Bands Supporting Long-Range Communications." Broadband Networks, Smart Grids and Climate Change. Springer New York, 2013. 215-241.

[20] J.W. Mayo, S. Wallsten. "Enabling efficient wireless communications: The role of secondary spectrum markets." Information Economics and Policy 22.1 (2010): 61-72.

[21] J.S. Warren, I.V. Hicks. "Combinatorial branch-and-bound for the maximum weight independent set problem." Technical report, Texas A&M University, 2006.

[22] Q. Zhao, B.M. Sadler. "A survey of dynamic spectrum access." Signal Processing Magazine, IEEE 24.3 (2007): 79-89.

[23] A. Kothari, D.C. Parkes, S. Suri. "Approximately-strategyproof and tractable multiunit auctions." Decision Support Systems 39.1 (2005): 105-121.

[24] R.B. Myerson. "Optimal auction design." Mathematics of operations research 6.1 (1981): 58-73.

[25] A.A. Ageev. "On finding critical independent and vertex sets." SIAM Journal on Discrete Mathematics 7.2 (1994): 293-295.

[26] V.E. Alekseev, V.V. Lozin. "Independent sets of maximum weight in (p, q)-colorable graphs." Discrete Mathematics 265.1 (2003): 351-356.

[27] L. Babel, G. Tinhofer. "A branch and bound algorithm for the maximum clique problem." Zeitschrift fur Operations Research 34.3 (1990): 207-217.

[28] E. Balas, C.S. Yu. "Finding a maximum clique in an arbitrary graph." SIAM Journal on Computing 15.4 (1986): 1054-1068.

[29] E. Balas, J. Xue. "Minimum weighted coloring of triangulated graphs, with application to maximum weight vertex packing and clique finding in arbitrary graphs." SIAM Journal on Computing 20.2 (1991): 209-221.

[30] Balinski, M. L. "Notes-on a selection problem." Management Science 17.3 (1970): 230-231.

[31] M.L. Basavaraju, S. Chandran, T. Karthick. "Maximum weight independent sets in hole-and dart-free graphs." Discrete Applied Mathematics 160.16 (2012): 2364-2369.

[32] R. Boppana, M.M. Halldorsson. "Approximating maximum independent sets by excluding subgraphs." BIT Numerical Mathematics 32.2 (1992): 180-196.

[33] A. Brandstadt, S. Mahfud. “Maximum weight stable set on graphs without claw and co-claw (and similar graph classes) can be solved in linear time.” Information Processing Letters 84.5 (2002): 251-259.

[34] A. Brandstandt “(P5,diamond)-free graphs revisited: structure and linear time optimization.” Discrete Applied Mathematics 138 (2004) 13-27.

[35] S. Butenko “Maximum independent set and related problems, with applications.” Diss. University of Florida, 2003.

[36] J. Pattillo, S. Butenko. ”Clique, independent set, and graph coloring.” Encyclopedia of Operations Research and Management Science (2011): 3150-3163.

[37] J. Clausen “Branch and bound algorithms-principles and examples.” Department of Computer Science, University of Copenhagen (1999): 1-30.

[38] R. Carraghan, P.M. Pardalos.“An exact algorithm for the maximum clique problem.” Operations Research Letters 9.6 (1990): 375-382.

[39] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition for cographs, SIAM J. Comput. 14 (1985) 926?934.

[40] F. Eisenbrand, S. Funke, N. Garg, J. Konemann. “A Combinatorial Algorithm for Computing a Maximum Independent Set in a t-Perfect Graph.” Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2003

[41] U. Feige. “Approximating maximum clique by removing subgraphs.” SIAM Journal on Discrete Mathematics 18.2 (2004): 219-225.

[42] T.A. Feo, M.G.C Resende, S.H. Smith. “A greedy randomized adaptive search procedure for maximum independent set.” Operations Research 42.5 (1994): 860-878.

[43] F. Gavril. “Algorithms on circular arc graphs.” Networks 4.4 (1974): 357-369.

[44] F. Gavril. "Algorithms for a maximum clique and a maximum independent set of a circle graph." Networks 3.3 (1973): 261-273.

[45] M. Grotschel, L. Lovasz, A. Schrijver. "Polynomial algorithms for perfect graphs." North-Holland mathematics studies 88 (1984): 325-356.

[46] M. Hifi "A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems." Journal of the Operational Research Society 48.6 (1997): 612-622.

[47] W.L. Hsu, I. Yoshiro, G.L. Nemhauser. "A polynomial algorithm for maximum weighted vertex packings on graphs without long odd cycles." Mathematical Programming 20.1 (1981): 225-232.

[48] M.J. Jou, G.J. Chang. "Survey on counting maximal independent sets." Proceedings of the Second Asian Mathematical Conference, S. Tangmance and E. Schulz eds., World Scientific, Singapore. 1995.

[49] T. Karthick. "On atomic structure of P5-free subclasses and Maximum Weight Independent Set problem." Theoretical Computer Science 516 (2014): 78-85.

[50] D. Kumlander. "A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search." Proc. 5th Int?l Conf. on Modelling, Computation and Optimization in Information Systems and Management Sciences. 2004.

[51] V.V. Lozin, M.Milani. "A polynomial algorithm to find an independent set of maximum weight in a fork-free graph." Journal of Discrete Algorithms 6.4 (2008): 595-604.

[52] Lozin, Vadim, R. Mosca. "Maximum independent sets in subclasses of P5-free graphs." Information Processing Letters 109.6 (2009): 319-324.

[53] Minty, George J. "On maximal independent sets of vertices in claw-free graphs." Journal of Combinatorial Theory, Series B 28.3 (1980): 284-304.

[54] R. Mosca. "Maximum weight independent sets in (P6, co-banner)-free graphs." Information Processing Letters 113.3 (2013): 89-93.

[55] Ostergard, R.J. Patric. "A fast algorithm for the maximum clique problem." Discrete Applied Mathematics 120.1 (2002): 197-207.

[56] W. Pullan. "Approximating the maximum vertex/edge weighted clique using local search." Journal of Heuristics 14.2 (2008): 117-134.

[57] W.R. Pulleyblank, F.B. Shepherd. "Formulations for the Stable Set Polytope" Proceedings Third IPCO Conference (1993)

[58] S. Sakai, M. Togasaki, K. Yamazaki. "A note on greedy algorithms for the maximum weighted independent set problem." Discrete Applied Mathematics 126.2 (2003): 313-322.

[59] S. Sanghavi, D. Shah, A.S. Willsky. "Message passing for maximum weight independent set." Information Theory, IEEE Transactions on 55.11 (2009): 4822-4834.

[60] N.Z. Shor. "Dual quadratic estimates in polynomial and Boolean programming." Annals of Operations Research 25.1 (1990): 163-168.

[61] G.J. Woeginger "Exact algorithms for NP-hard problems: A survey." Combinatorial Optimization?Eureka, You Shrink!. Springer Berlin Heidelberg, 2003. 185-207.

[62] Q. Wu, J.K. Hao, and F.Glover. "Multi-neighborhood tabu search for the maximum weight clique problem." Annals of Operations Research 196.1 (2012): 611-634.

[63] V.V Lozin, M.Milanic. "A polynomial algorithm to find an independent set of maximum weight in a fork-free graph." Journal of Discrete Algorithms 6.4 (2008): 595-604.

[64] C.Q. Zhang "Finding critical independent sets and critical vertex subsets are polynomial problems." SIAM Journal on Discrete Mathematics 3.3 (1990): 431-438.