A Parallel Algorithm for Maximal Independent Set

Edward Duong - *edwardduong@cmail.carleton.ca*

December 5, 2015

Graph theory is a diverse and wide topic with numerous problems, many of which lend itself to be solvable using parallel algorithms. One such problem is the minimal independent set (MIS). We discuss the basic concepts of the inimal independent set and the algorithm that can construct a MIS in parallel.

The idea of independent sets have been well studied in the past decades. Discussion on this topic can be traced back to as early as the 1960. During that era Moon et al.[9] and Erdos[4] provided bounds on the size of maximal cliques in random graphs. (We will see later that there is a direct relationship between the maximal clique and the maximal independent set of a graph.)

Today, there is still quite a lot of research on the topic of independent sets. In recent years, the focus is on parallel and distributed MIS algorithms in the form of approaching the problem with different techniques, such has greedy algorithms[1] or by approximation[3].

1 Maximal Independent Set

An independent set S for a graph $G = (V, E)$ is a subset of vertices in V such that no two vertices in S are adjacent. That is to say, there are no edges in E that connect any vertex in S with any other vertex in S.

A maximal independent set (MIS) is an independent set that is not a subset of any other indpendent set of G. If given an an independent set that is not maximal, one can construct a MIS from it. This is done by continuously adding vertices to the independent set (without violating the independent set property) until no more vertices can be added. A graph may have multiple independent sets and multiple MISs. An example follows in Figure 1.

Definition 1 *Given a graph $G = (V, E)$, a maximal independent set $U \subset V$ satisfies the conditions that $N(U) \cap U \neq \emptyset$ and $N(U) \cup U = V$, where $N(U)$ denotes the vertex neighbors of the set U.*

The MIS with the highest cardinality, i.e. most number of vertices, is the **maximum** independent set. Again, there may be many maximum independent sets if they share the same cardinality. Note that a maximum independent set is maximal and hence it is a MIS, but not the converse is not always true.

2 Relationships with Other Graph Problems

The MIS of a graph G shares an interesting relationship with the clique problem. It is purportedly so well-known that it is not cited in discussions - leaving the origin of discovery somewhat lost.

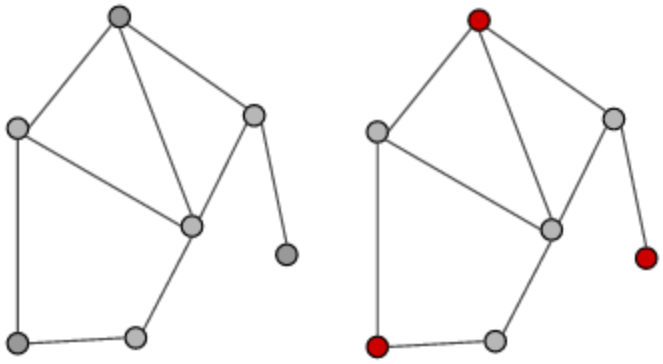

Figure 1: On the right, one of many minimal independent sets (in red) of the graph on the left

A clique of G is a subset that forms a complete graph (an exampple is shown in figure 2). A complete graph is a simple undirected graph where each distinct pair of vertices are connected by an edge. The relationship it holds with a MIS can be seen by taking $\overline{G}$, the edge complement of G. Intuitively, we observe why a MIS in G gives a maximal clique in $\overline{G}$. In $\overline{G}$, there is an edge between a pair of vertices that formly did not have an edge in G. A MIS in G guarantees that all vertices selected do not have connecting edges. Hence, in $\overline{G}$ they must be connected.

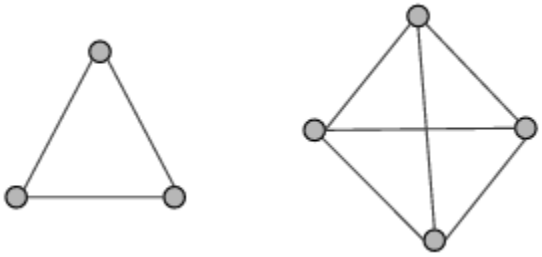

Figure 2: Left: complete graph K_3. Right: complete graph K_4

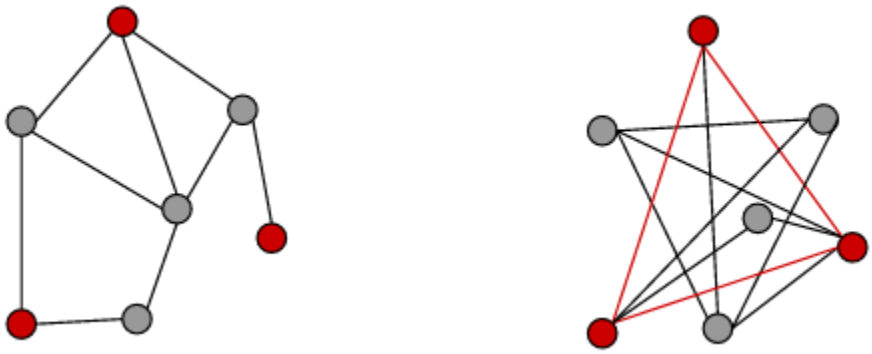

Figure 3: Left: a MIS. Right: Clique representation of graph complement.

Another well-known relationship is the MIS to minimal vertex cover problem.[6] A minimal vertex cover of $G = (V, E)$ is a subset of vertices $C \subset V$ such that every edge $e \in G$ has at least 1 of its endpoint is in C. Taking the vertex set difference $V \setminus S = C$, where S is a MIS, gives a minimal vertex cover C. It also follows that a vertex set is a vertex cover if and only if its complement is an indepdent set.

It is a bit less obvious why this is the case, but we can look closer at a MIS property to see why. A vertex not in the MIS S has at least one neighbor in S. Each vertex not in the MIS S becomes the minimal vertex cover, hence the cover can reach all vertices not in the cover, so there must be an edge that connects them.

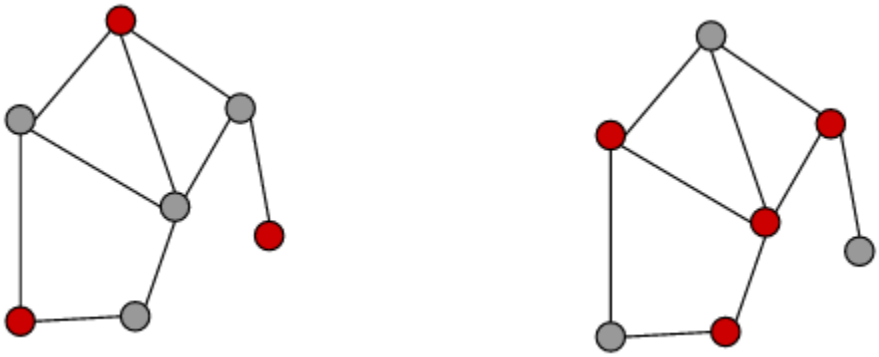

Figure 4: Left: a MIS. Right: Vertex cover from the MIS complement

3 Problem Types of Independent Sets

It is worth making a distinction between solving a single MIS, solving all MISs and solving the maximum independent set of a graph. Surprisingly, the difficulty between solving the first versus the last two problems is monumental. A single MIS of a graph is achievable in $O(n)$ time. The complexity for computing all MISs varies on the graph structure; however, it is know that a general graph with n vertices has at most $3^{n/3}$ maximal independent sets[9]. By repeating the linear MIS solution, we get a $O(3^{n/3})$ bound for outputing all MISs. Lastly, solving a maximum independent set is NP-hard. Current leading algorithms[11][2][5], from as recent as 2010, achieve bounds between $O(1.2108^n)$ and $O(1.2209^n)$.

4 MIS Algorithms

Below, we present a sequential algorithm to output a single MIS of a graph.

During each iteration, one new vertex is added to S. At all times in the execution, S is always an independent set. This is guaranteed by the step in which neighbors of v are removed so that they can never be selected for S in future iterations.

The runtime complexity is linear time. An iteration removes at least one vertex, but possibly more if it has neighbors. We know that a vertex can only be removed once and a removed vertex is never added back to V. Therefore the algorithm removes exactly $|V|$ vertices. The algorithm is guaranteed to terminate because at least one vertex is removed each iteration.

Algorithm 1 Sequential MIS

```
1: procedure MIS(G = (V, E))
2:     S = ∅
3:     while V ≠ 0 do
4:         Arbitrarily select v ∈ V
5:         Add v to S
6:         Remove v and its neighbors from V
       return S
```

We now present a modern variant[8][10] of the prominent parallel MIS algorithm by Luby[7], which was discovered in 1985. Both are very similar and use probabilitic techniques to achieve a logarithmic runtime in a parallel setting. However, the modern variant is 'relatively' easier to prove.

Algorithm 2 Parallel MIS

```
1: procedure MIS(G = (V, E))
2:     S = ∅
3:     while V ≠ 0 do
4:         Each v ⊆ V selects a random number r(v) in [0, 1] and sends it to its neighbors, w ∈
   N(v)
5:         If r(v) < r(w), v informs its neighbors and adds itself to S
6:         If v is added, remove v and N(v) from V
       return S
```

We begin by highlighting properties of Algorithm 2, starting with showing that the algorithm terminates. In each iteration, a vertex is assigned the lowest random number $r(v)$ (ties are assumed to be broken arbitrarily by lexicographical name, etc.) This vertex is removed from V and is never re-added back to V. Therefore $|V|$ is decreases with each iteration and the algorithm must eventually terminate.

Similar to the sequential algorithm, S always represents an independent set throughout execution. Again, this is guaranteed by the removal of $N(v)$.

To show that on expectation a logarithmic number of iterations are required, we show that the number of edges removed is some constant fraction of $|E|$. An outline of the proof is as follows:

1. Transform undirected edges of G into an equivalent pair of directed edges
2. Define a probabilistic event where a vertex and its neighbors are removed, determine its probability and the number of edges removed should it occur
3. Using Linearity of Exepected Value, i.e. $E[\sum_i X_i] = \sum_i E[X_i]$ where X_i represents a random variable, we show that the event we defined when applied over all edges (ordered pairs ofdirected edges) has expected value $|E|/2$, meaning that in expectation at least half of the edges are removed each iteration

4.1 Part 1

For each undirected edge that connects (vw) in G we replace it with 2 directed edges (v, w) and (w, v) for presentation. This is done for reasons that will become apparent later in the proof. From now on, when refering to $d(v)$, we refer to the outgoing degree of a vertex v (which is the same as the degree in the undirected graph).

4.2 Part 2

When v is added to the independent set S and is removed from V, along with its neighbors, the condition $r(v) < r(w), w \in N(v)$ must be true (as stated in the algorithm). Also, $r(v) < r(x), x \in N(w)$ otherwise it is possible that x would have been added to the independent set in place of v. We label this event $(v- > w)$ (v preemptively removes a neighbor w). The probability of the event is at least $1/(d(v) + d(w))$, the maximum number of vertices adjacent to v, w or both v and w. Clearly some edges are possibly doubly counted and leads to a lower probability than in actual, but the lower bound is sufficient and necessary for the proof. Next, we ask how many edges are removed when event $(v- > w)$ occurs? At least all directed edges (wx), which there are at least $d(w)$ many. We need to consider the event $(w- > v)$ as well since both occur together in the undirected graph. The probability of $(w- >)$ is also at least $1/(d(v) + d(w))$ and the edges removed is at least $d(v)$.

4.3 Part 3

Let the event $(v- > w)$ that adds v to the independent set and removes $d(w)$ edges be represented by the random variable $X_{(v->w)}$. Similarly, let the event $(w- > v)$ be represented by the random variable $X_{(w->v)}$.

$$X_{(v->w)} = \begin{cases} 0 & \textit{if event does not occur} \\ d(w) & \textit{if event occurs} \end{cases}$$

$$X_{(w->v)} = \begin{cases} 0 & \textit{if event does not occur} \\ d(v) & \textit{if event occurs} \end{cases}$$

In the undirected graph, an edge (v, w) is represented by two edges and two such random variables: $X_{(v->w)}$ and $X_{(w->v)}$

Using the Linearity of Exepected Value, we determine the expected value over all edges:

$$E[\sum_{\{v,w\}\in E} X_{\{v,w\}}] = \sum_{\{v,w\}\in E} E[X_{\{v,w\}}] + E[X_{\{w,v\}}] \tag{1}$$

$$= \sum_{\{v,w\}\in E} d(w)\cdot Pr(Event(v->w)) + d(v)\cdot Pr(Event(w->v)) \tag{2}$$

$$= \sum_{\{v,w\}\in E} d(w)\cdot \frac{1}{d(v)+d(w)} + d(v)\cdot \frac{1}{d(w)+d(v)} \tag{3}$$

$$= \sum_{\{v,w\}\in E} \frac{d(w)}{d(v)+d(w)} + \frac{d(v)}{d(w)+d(v)} \tag{4}$$

$$= \sum_{\{v,w\}\in E} \frac{d(v)+d(w)}{d(v)+d(w)} \tag{5}$$

$$= \sum_{\{v,w\}\in E} 1 \tag{6}$$

$$= |E| \tag{7}$$

In expectation $|E|$ directed edges are removed in each iteration. So in expectation $|E|/2$ undirected edges are removed. Therefore, the algorithm is expected to output a MIS in $O(logn)$ steps.

References

[1] Guy E Blelloch, Jeremy T Fineman, and Julian Shun. Greedy sequential maximal independent set and matching are parallel on average. In *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*, pages 308–317. ACM, 2012.

[2] Nicolas Bourgeois, Bruno Escoffier, Vangelis Th Paschos, and Johan MM van Rooij. A bottom-up method and fast algorithms for max independent set. In *Algorithm Theory-SWAT 2010*, pages 62–73. Springer, 2010.

[3] Mostafa H Dahshan. Maximum independent set approximation based on bellman-ford algorithm. *Arabian Journal for Science and Engineering*, 39(10):7003–7011, 2014.

[4] P Erdös. On cliques in graphs. *Israel Journal of Mathematics*, 4(4):233–234, 1966.

[5] Fedor V Fomin, Fabrizio Grandoni, and Dieter Kratsch. A measure & conquer approach for the analysis of exact algorithms. *Journal of the ACM (JACM)*, 56(5):25, 2009.

[6] Gallai. ber extreme punkt- und kantenmengen. *Etvs Sect. Math*, 1959.

[7] Michael Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 1–10. ACM, 1985.

[8] Yves Métivier, John Michael Robson, Nasser Saheb-Djahromi, and Akka Zemmari. An optimal bit complexity randomized distributed mis algorithm. *Distributed Computing*, 23(5-6):331–340, 2011.

[9] John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965.

[10] Wattenhofer R. Maximal independent set. *ETH Zurich*.

[11] John Michael Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.