Proving NP Completeness of the Linear complementarity problem via reduction from Independent Set

Mihir Kulkarni

January 15, 2015

Abstract

Given a graph G and an integer k, the Independent Set problem (decision version) asks the question:
Does there exist an independent set of size $\geq k$ in G ?
We reduce the decision version of the Independent Set Problem (IS) to an Integer Linear Program(ILP). Then, we formulate an instance of the Linear Complementarity Problem in polynomial time. A solution for this will solve the ILP and therefore, the LCP. Doing this for all values of k will give us the maximum Independent Set.

1 Reducing from Independent Set(IS) to Integer Linear Program (ILP)

We reduce to ILP using the techniques outlined in `http://yaroslavvb.blogspot.in/2011/03/linear-programming-fo` `html`
Given a graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, we create

1. A variable x_i for each vertex, constrained to be in $\{0, 1\}$.

2. Constraints $x_k + x_j \leq 1 \ \forall \ (x_j, x_k) \in E$ where E is the edge set.

3. The linear function $\sum_{i=1}^{n} x_i$ to be maximized.

A solution for this in 1-0 gives us the maximum independent set.

2 ILP to LCP

We use a idea similar to Chung's paper.
The constraints are of the form $x_k + x_j \leq 1 \forall (x_j, x_k) \in E$ where E is the edge set.
Let $freq\,(v)$ be the number of edges that vertex v appears in. Let
$2 \sum_{v \in V} freq\,(v) + 2n = SumV$ Hence, $SumV = 2 \times 2 \times m + 2n = 4m + 2n$. This is because each edge is counted twice in the sum, once for each of its endpoints
For our LCP, we take the order of matrix M $= 2m + 2 + n + SumV = 6m + 2 + 3n$.
We formulate the LCP as

2.1 The vector $\vec{x}$

$\vec{x} = \begin{bmatrix} x_{j0} & x_{k0} & \cdots & x_{j(m-1)} & x_{k(m-1)} & v_0 \cdots & v_{n-1} & z_0 & \cdots & z_{SumV-1} \end{bmatrix}$
Here,

1. (x_{jl}, x_{kl}) represent the $l + 1^{th}$ edge

2. v_i represents the $i + 1^{th}$ vertex.

3. Values of z_i are unimportant. They are present to ensure that the dimension of $\vec{x}$ is the same as M.

We have $2m$ constraints for the edges- one for each vertex for each edge.
There are n constraints for each vertex.
The other $SumV = 4m + 2n$ constraints force the equality of the repeating vertices.

2.2 The matrix M

$$M = \left[\begin{array}{ccccccccccc}
-1 & -1\cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
-1 & -1\cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0\cdots & -1 & -1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0\cdots & -1 & -1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \ddots & & & & & & & & & \\
0 & 0\cdots & 0 & 0 & \cdots & -1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0\cdots & 0 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 \\
0 & 0\cdots & 0 & 0 & \cdots & 0 & 0 & -1 & \cdots & 0 & 0 \\
\vdots & \ddots & & & & & & & & & \\
0 & 0\cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & -1 & 0 \\
1 & 0\cdots & 0 & 0 & \cdots & -1 & 0 & 0 & \cdots & 0 & 0 \\
-1 & 0\cdots & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0\cdots & 1 & 0 & \cdots & -1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0\cdots & -1 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \ddots & & & & & & & & & \\
0 & 0\cdots & 0 & 0 & \cdots & -1 & -1 & -1 & \cdots & -1 & 1 \\
0 & 0\cdots & 0 & 0 & \cdots & 1 & 1 & 1 & \cdots & 1 & -1
\end{array}\right]
\begin{array}{l}
\left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} 2m\ Edge\ Constraints \\
\left.\begin{array}{l} \\ \\ \\ \\ \\ \end{array}\right\} n\ Vertex\ 0-1\ constraints \\
\left.\begin{array}{l} \\ \\ \\ \\ \\ \end{array}\right\} SumV\ Vertex\ equality\ constraints \\
\left.\begin{array}{l} \\ \\ \end{array}\right\} Enforcing \sum_{i=0}^{n-1} v_i = b
\end{array}$$

$$\underbrace{\hphantom{-1\quad -1\cdots\quad 0\quad 0}}_{Edges} \qquad \underbrace{\hphantom{-1\quad -1\quad -1\quad \cdots\quad -1}}_{Vertices} \quad \underbrace{\hphantom{-1}}_{b}$$

2.3 The vector $\vec{q}$

For $\vec{w} = M\vec{x} + \vec{q}$, with 1 as the vector with all 1's, $\begin{bmatrix}1^{2m} & 1^n & 0^{SumV} & b & -b\end{bmatrix}$

2.4 Coordinates 1 to 2m - edge constraints

For the $l + 1^{th}$ edge, we encode the constraints
$w_{2l} = w_{2l+1} = 1 - (x_{kl} + x_{jl})$
Therefore $w_i = 1 - (x_k + x_j) \forall i \in \{1 \ldots 2m\}$, for some edge $(x_k, x_j) \in E$
Each edge therefore gives two constraints.
Note that non-negativity of w_i forces $x_k + x_j \leq 1 \forall (x_j, x_k) \in E$

2.5 Constraining the vertices to be only 0-1

Constraints $2m + 1$ to $2m + n$ are solely for the purpose of constraining the variables to be in {0,1}.
We create the following conditions:
$\forall i \in \{2m + 3, \ldots, 2m + 2 + n\}$
$q_i = 1$
$w_i = 1 - v_{i-(2m+3)}$

2.6 Equality constraints - coordinates greater than 2m+2 + n

We now force equality of the v_js that represent the same vertex.

Given a vertex v_j, recall that $freq(v_j)$ represents the number of edges it appears in and relabelling the variable v_j as $x_{j_{freq(v_j)}}$ (to force its equality with all the others), we let x_{j_l} represent the l^{th} occurence of the vertex v_j, for $0 \leq l \leq freq(v_j)$

We encode the constraints with the following equations:
$q_{j_l j_{l+1} 1} = q_{j_l j_{l+1} 2} = 0$
$w_{j_l j_{l+1} 1} = x_{j_l} - x_{j_{l+1}}$
$w_{j_l j_{l+1} 2} = x_{j_{l+1}} - x_{j_l}$
$\forall 0 \leq l \leq freq(v_j)$

Non-negativity of these expressions forces $x_{j_l} = x_{j_{l+1}}$ and $w_{j_l j_{l+1} 1} = w_{j_l j_{l+1} 2} = 0$

Each v_j appears in two equations and the vertex it represents gives $2freq\,(v_j)$ equations. Summing these for all v_j gives us the number of such constraints as: $2\sum_{x\in V} freq\,(x) + 2n = SumV$

2.7 Setting sum of vertex variables as b

The last two equations are :

$w_{6m+3n+1} = -\sum\limits_{i=0}^{n-1} v_i + b$

$w_{6m+3n+2} = \sum\limits_{i=0}^{n-1} v_i - b$

As was idea used in Chung's paper, non-negativity of w forces $w_{6m+3n+1} = w_{6m+3n+2} = 0$, which we get by adding these two equations.

This gives us $\sum\limits_{i=0}^{n-1} v_i = b$

2.8 Correctness

We have the constraint $w_i x_i = 0$

2.8.1 $i > 2m + n$

$w_i x_i = 0$ holds trivially because $w_i = 0$.

2.8.2 $2m + 1 \leq i \leq 2m + n$

Let $l = i - (2m + 3)$
$w_i = (1 - v_l)$
$w_i x_i = (1 - v_l)v_l = 0$
This forces $v_l = 0$ or $v_l = 1$

2.8.3 $i \leq 2m + 2$

$\forall (x_j, x_k) \in E$ we have the following constraints:
$(1 - (x_k + x_j))x_k = 0$
$(1 - (x_k + x_j))x_j = 0$

Case 1 $x_k = x_j = 0$ (both vertices from corresponding edge is not included) Both constraints are satisfied.

Case 2 Exactly one of x_j, x_k is 0. Wlog, assume this is x_j. This forces $1 - x_k - x_j = 0$ or $x_k = 1$ (We pick x_k in the independent set).

Case 3 Both are non-zero. This is not possible as it would imply that $w_i = -1$ (x_j, x_k are either 0 or 1 because they are equal to some $v_i \in \{0, 1\}$). Hence, only one gets picked.
Therefore only one vertex is picked from each edge. Moreover, by constraints encoded in 2.6, all x_js representing a variable v_j must be equal to v_j.
The above conditions ensure that the v_js picked (i.e., set to 1) form an independent set. Given a size b, we have $\sum\limits_{i=0}^{n-1} v_i = b$, i.e. the vertices picked form an independent set of size b, if it exists.

2.9 Algorithm and Reduction

We have shown that solution satisfies all the constraints of the ILP and have therefore shown a polynomial time reduction for the ILP.
Given a polynomial time algorithm for LCP, we solve this LCP for all values of b from 1 to n. The maximum value of b for which we get a solution gives us the size of the maximum independent set.

Hence, a solution for this LCP solves the corresponding ILP and hence, the original Independent Set problem, given by $(v_i)_{0\leq i\leq n-1}$ in polynomial time.

3 Alternate Formulation

We can formulate this as a linear program with complementarity constraints as:

$LPLCC(G)$
$max 1^T v = \sum_{i=1}^{n} v_i$
$v \in \mathrm{R}^n$
$v \geq 0$
$w = Av - 1 \geq 0$
$v_i w_i = 0 \forall i$
where $(Av)_i = \sum_{v \in ClosedNbr(v_i)} v$ where $ClosedNbr\,(v_i)$ is the set containing v_i and all its neighbours
We need to show that this relaxation gives an optimal value which is equal to the size of the maximum independent set.

3.1 Proof

If a vertex v_i is nonzero, then $ClosedNbr\,(v_i)$ gives a contribution of exactly 1. This is because $\left(\sum_{v \in ClosedNbr(v_i)} v - 1\right) v_i = 0 \forall i$

Using this, we now claim that the optimization problem is nothing but finding a partition of vertices such that each set in the partition is $ClosedNbr\,(v_i)$ for some vertex v- of the maximum size. The number of sets gives us the optimal value.

But this indeed, is exactly what the maximum independent set is- we have that each vertex in the set has none of its neighbours in the set and the union of $ClosedNbr\,(v_i)$ for each vertex v_i in the set is the entire graph. Therefore a solution to this gives us the size of the maximum independent set.

3.2 Proof by induction- first version

Another way of saying this is by proving the following theorem:

Theorem 1. *There always exists a 0-1 optimal solution*

Proof. We prove the theorem via strong induction on n.

1. **Base case:** Trivially true for the singleton graph and graph with two vertices.

2. **Inductive case:** Consider a solution $v1$ that assigns a nonzero value to some v_i. Then, we have that $\sum_{x \in ClosedNbr(v_i)} x = 1$.

   Now consider a solution $v2$ with
   $v_i = 1$
   $x = 0\ \forall x \in ClosedNbr\,(v_i)$ We show that this can be used to create an optimal 0-1 solution.

   Consider $V' = V - ClosedNbr(v_i)$ and $E' = \{(v_i, v_j) \colon v_i, v_j \in V$ and $v_i, v_j \in E\}$
   Let $G' = (V', E')$ and A', v' represent the matrix and vector obtained after removing entries corresponding to $ClosedNbr\,(v_i)$ from A and v. Let $v1', v2'$ represent the corresponding solutions obtained from $v1$ and $v2$.

Note that once we set values of $v \in ClosedNbr(v_i)$ in $v1$, the problem reduces to finding $max 1^T v'$ subject to all the earlier constraints, and also the additional constraints $\sum_{y \in ClosedNbr(x)} y = 1 \; \forall x \in ClosedNbr\,(v_i)\,| x \neq 0$.

On the other hand, if we set the values as were set in $v2$, then the problem reduces only to finding $max 1^T v'$ *without the additional constraints.* This is $LPLCC(G')$.

This implies $max\ 1^T v1' \leq max\ 1^T v2'$. As $ClosedNbr(v_i)$ gives the same contribution (1) in both cases, we have $max\ 1^T v1 \leq max\ 1^T v2$. Hence, there exists an optimal $v2$ if $v1$ is optimal.

By applying inductive hypothesis on $LPLCC(G')$, an optimal 0-1 solution $v2'$ exists for G'. Using this and the fact that
$v_i = 1$
$x = 0 \; \forall x \in ClosedNbr\,(v_i)$

□

3.3 Proof by induction- second version

Another way of saying this is by proving the following theorem:

Theorem 2. *There always exists a 0-1 optimal solution*

Proof. We prove the theorem via strong induction on n.

1. **Base case:** Trivially true for the singleton graph and graph with two vertices.

2. **Inductive case:** Consider a solution $v1$ that assigns a nonzero value to some v_i. Then, we have that $\sum_{x \in ClosedNbr(v_i)} x = 1$.

   Now consider a solution $v2$ with
   $v_i = 1$
   $x = 0 \; \forall x \in ClosedNbr\,(v_i)$ We show that this can be used to create an optimal 0-1 solution.

   Consider $V' = V - ClosedNbr(v_i)$ and $E' = \{(v_i, v_j) \colon v_i, v_j \in V' \text{ and } (v_i, v_j) \in E\}$
   Let $G' = (V', E')$ and A', v' represent the matrix and vector obtained after removing entries corresponding to $ClosedNbr\,(v_i)$ from A and v. Let $v1', v2'$ represent the corresponding solutions obtained from $v1$ and $v2$.

   Let $opt(v')$ refer to an optimal solution of G' and $opt(v)$ refer to one of G.

   Lemma 1. *If $v_i \neq 0$ in $opt(v)$, then $\sum_{x \in ClosedNbr(v_i)} x + 1^T opt(v') = 1 + 1^T opt(v') = 1^T opt(v)$*

   Proof. If v_i is nonzero, then $ClosedNbr(v_i)$ gives a contribution of 1. The value to be maximized can be rewritten as
   $\sum_{x \in ClosedNbr(v_i)} x + 1^T v'$
   Now E can be partitioned into 3 subsets.

   (a) $E1$: This consists of all edges of G'

   (b) $E2$: This contains all the edges connecting v_i to its neighbours.

   (c) $E3$: This contains all the remaining edges, i.e. the edges connecting vertices in $ClosedNbr\,(v_i)$ which are not v_i to vertices outside $ClosedNbr\,(v_i)$.

   Now, the optimal value for the $LPLCC(G')$ is given by $1^T opt(v')$. This LPLCC is solved by using only the edge constraints from the set $E1$. Given v_i is non-zero, $\sum_{x \in ClosedNbr(v_i)} x = 1$.
   Consider the graph given by $G'' = (V, E1 \cup E2)$ with the additional constraint that $v_i \neq 0$.
   Let $V(E'')$ denote the set of all vertices that appears in an edge set E''. The solution for $LPLCC(G'')$ can be obtained by solving $LPLCC(G')$ and adding 1 to it, because G'' is obtained by putting together the disjoint graphs $G'(V(E1), E1)$ and $G'''(ClosedNbr\,(v_i)\,, E2)$ that have no

common edges or vertices; and $\sum_{x \in ClosedNbr(v_i)} x = 1$.
The optimal solution for G'' will necessarily be at least as large as that as one for G which has $v_i \neq 0$, because G'' is obtained from G by removing some edges (constraints).
However, no solution can give a value strictly greater than $1^T opt(v)$, because $opt(v)$ is optimal. Therefore, the two values must be equal. $\square$

Consider an optimal solution $v1$ for G that sets $v_i \neq 0$. Now consider G'. By inductive hypothesis, this has a 0-1 optimal solution. We also have that $\sum_{x \in ClosedNbr(v_i)} x = 1$.
We can construct a feasible 0-1 solution for G by setting

$v_i = 1$
$x = 0 \ \forall x \in ClosedNbr\,(v_i)$

and assigning values to vertices in G' equal to their values in its optimal 0-1 solution. This is indeed feasible because

(a) Vertices of G' satisfy constraints defined in $E1$ and those in $ClosedNbr\,(v_i)$ satisfy constraints of $E2$. Both sets of constraints deal with only one of these two disjoint sets.

(b) The constraints in $E3$ are satisfied because if any neighbour y of some $x \in ClosedNbr\,(v_i)$ is assigned 1, then all it's neighbours must be zero - because $\sum_{z \in ClosedNbr(y)} z = 1$ in $LPLCC(G')$, and any neighbour(s) from $ClosedNbr\,(v_i)$ is(are) also set to zero by the construction. The only other value such a vertex can take is 0, by inductive hypothesis.

By 2, this is in fact an optimal solution for G. Hence proved.

$\square$

3.4 Proof by induction- third version

Another way of saying this is by proving the following theorem:

Theorem 3. *There always exists a 0-1 optimal solution*

Proof. We prove the theorem via strong induction on n.

1. **Base case:** Trivially true for the singleton graph and graph with two vertices.

2. **Inductive case:**

   Consider G,G'' and G'. Let their optimal solutions be denoted by opt, opt'' and opt' respectively. If $LPLCC(G') = (G', A', v')$ then $LPLCC(G) = (G, A, v)$ and $LPLCC(G'') = (G'', A'', v'')$ where $A = \begin{bmatrix} A' & \delta \\ \delta^T & ClosedNbrEdge(v_n) \end{bmatrix}$ and $A'' = \begin{bmatrix} A' & 0 \\ 0 & ClosedNbrEdge(v_n) \end{bmatrix}$

   This is because δ corresponds to edges with one end in G' and another in $ClosedNbr(v_n)$. Symmetry causes the appearance of δ^T in A and the 0 blocks are because it is precisely these edges that vanish when we create G'' from G.

   Also, $v = \begin{bmatrix} v' & ClosedNbr\,(v_n) \\ ClosedNbr(v_n) & \end{bmatrix}$ by construction.

Lemma 2. *If $v_n \neq 0, opt' + 1 = opt''$*

Proof. $v_n \neq 0$ implies $\sum_{x \in ClosedNbr(v_n)} x = 1$. Also, $G'' = G' \cup ClosedNbr\,(v_i)$, where G' and $ClosedNbr\,(v_n)$ are disjoint from one another. Hence, we have that
$opt'' = opt(G' \cup ClosedNbr\,(v_n))$
$= opt' + opt(ClosedNbr\,(v_n))$
(because the two are disjoint) $= opt' + 1$ $\square$

In particular, a 0-1 solution opt' for G' gives a 0-1 solution opt'' for G'' obtained by setting
$v_n = 0 x = 0 \forall x \in ClosedNbr\,(v_n)$ and other variables set to their values in opt'

Lemma 3. *A 0-1 optimal solution represents a maximum independent set.*

Proof. Consider a 0-1 feasible solution. Let $v_i = 1$ represent picking a vertex in the set and $v_i = 0$ represent otherwise. The constraints $v_i(\sum_{x \in ClosedNbr(v_i)} -1) = 0$ force that if $v_i = 1$, then $x = 0 \forall x \in ClosedNbr\,(v_i)$. This is essentially the independent set restriction. Each non-negativity constraint for A is of the form $\sum_{x \in ClosedNbr(v_i)} \geq$ 1- meaning that we pick at least one vertex from each closed neighbourhood. This therefore represents a maximal independent set. The objective function $1^T v$ that is being maximized represents the size of the independent set. Therefore , if a 0-1 solution is optimal, it represents the maximum independent set. $\square$

Lemma 4. *$opt'' \geq opt$ if the solution is 0-1*

Proof. G'' represents the graph obtained from G by removing certain edges. Any independent set that picks v_n in G will therefore also be an independent set in G''. The lemma follows. $\square$

By induction hypothesis, a 0-1 optimal opt' exists. This means that there exists an optimal $opt'' = opt' + 1$, which is 0-1. However, this is feasible in G as well. This is because we do not pick any neighbour of v_n in this independent set, and therefore the additional edges we add to G'' to obtain G do not change the feasibility of the solution. Given that opt'' is feasible in G and opt'' is optimal in G'', the above lemma tells us that opt'' is a 0-1 optimal solution for G. $\square$