Maximum independent set (stable set) problem: Computational testing and a Satisfiability (3-SAT) model

Prabhu Manyem (retired)
College of Science
Nanchang Institute of Technology
Nanchang, China

M.I.S. Definition

The decision version of the maximum independent set (M.I.S.) problem:

- Given: A constant P and a graph $G = (V, E)$.
- Question: Is there a subset $S \subseteq V$ such that (i) no two members of S are adjacent to each other, and (ii) $|S| \geq P$?
- $n = |V|, \quad 1 \leq P$ (integer) $\leq n$.
- The decision version of M.I.S. is known to be NP-complete.

M.I.S. Definition

The decision version of the maximum independent set (M.I.S.) problem:

- Given: A constant P and a graph $G = (V, E)$.
- Question: Is there a subset $S \subseteq V$ such that (i) no two members of S are adjacent to each other, and (ii) $|S| \geq P$?
- $n = |V|, \quad 1 \leq P \text{ (integer)} \leq n$.
- The decision version of M.I.S. is known to be NP-complete.

0-1 Integer Program for M.I.S.

$$
\begin{aligned}
\text{Maximise } Z_1 &= \textstyle\sum_{j \in V} (F_j) \\
F_i + F_j &\leq 1 \;\; \forall\, (i, j) \in E \\
F_j &\in \{0, 1\} \;\; \forall\, j \in V.
\end{aligned} \tag{1}
$$

$F_j = 1$ if vertex $j \in$ (Independent set S), and zero otherwise.

Ind. Set $S = \{\, j \in V \mid F_j = 1\}$.

Binary Search

Integer solution: For every vertex $i \in V$, F_i is either zero or one.

Binary Search

Integer solution: For every vertex $i \in V$, F_i is either zero or one.

Binary search approach: Do a binary search on the value of k for the following problem:

$$\begin{aligned} F_i + F_j &\leq 1 \ \ \forall\, (i,j) \in E \\ F_j &\in \{0,1\} \ \ \forall\, j \in V \\ \textstyle\sum_{j \in V}(F_j) &= k. \end{aligned} \tag{2}$$

(Is there an independent set in G of size k?)

$k = 1,\ 2,\ 4,\ 8,\ \cdots,\ n$.

Binary Search

Integer solution: For every vertex $i \in V$, F_i is either zero or one.

Binary search approach: Do a binary search on the value of k for the following problem:

$$\begin{aligned} F_i + F_j &\leq 1 \;\; \forall\, (i,j) \in E \\ F_j &\in \{0,1\} \;\; \forall\, j \in V \\ \textstyle\sum_{j\in V}(F_j) &= k. \end{aligned} \tag{2}$$

(Is there an independent set in G of size k?)

$k = 1, 2, 4, 8, \cdots, n$.

Linear relaxation:

$$\begin{aligned} F_i + F_j &\leq 1 \;\; \forall\, (i,j) \in E \\ \forall\, j \in V,\; 0 \leq F_j &\leq 1 \\ \textstyle\sum_{j\in V}(F_j) &= k. \end{aligned} \tag{3}$$

How to find an integer solution to the Linear relaxation?

Approach 1: Valid Inequalities

- Clique inequalities: For a clique of size k, add:

  $x_{c1} + x_{c2} + \cdots + x_{ck} \leq 1.$

- Cycle inequalities: For a cycle of size k, add:

  $x_{c1} + x_{c2} + \cdots + x_{ck} \leq \lfloor k/2 \rfloor.$

Approach 1: Valid Inequalities

- Clique inequalities: For a clique of size k, add:

  $x_{c1} + x_{c2} + \cdots + x_{ck} \leq 1.$

- Cycle inequalities: For a cycle of size k, add:

  $x_{c1} + x_{c2} + \cdots + x_{ck} \leq \lfloor k/2 \rfloor.$

- Adding Valid Inequalities improved the rate at which integer solutions were obtained.

Approach 2: Find an integer solution to the Linear relaxation

Find a function $g(x)$ such that an integer solution is the least cost solution to the Linear relaxation.

Approach 2: Find an integer solution to the Linear relaxation

Find a function $g(x)$ such that an integer solution is the least cost solution to the Linear relaxation.

Linear relaxation:

$x_i + x_j \leq 1 \ \forall \ (i, j) \in E, \quad 0 \leq x_j \leq 1 \ (\forall \ j \in V), \quad \sum_{j \in V}(x_j) \ = \ k.$

Every vertex is a bin. Amount in bin $j = x_j$.

Function value for bin $j = g(x_j)$.

Minimise $Z = \sum_{j \in V} \ g(x_j)$.

$X = (x_1, x_2, \cdots, x_n)$.

Approach 2: Find an integer solution to the Linear relaxation

Find a function $g(x)$ such that an integer solution is the least cost solution to the Linear relaxation.

Linear relaxation:

$x_i + x_j \leq 1 \; \forall \; (i,j) \in E, \quad 0 \leq x_j \leq 1 \; (\forall \; j \in V), \quad \sum_{j \in V}(x_j) \; = \; k.$

Every vertex is a bin. Amount in bin $j = x_j$.

Function value for bin $j = g(x_j)$.

Minimise $Z = \sum_{j \in V} \; g(x_j)$.

$X = (x_1, x_2, \cdots, x_n)$.

(Needed) If for solution X, the function g achieves its minimum, then X should be an integer solution.

Approach 2: Find an integer solution to the Linear relaxation

Find a function $g(x)$ such that an integer solution is the least cost solution to the Linear relaxation.

Linear relaxation:

$x_i + x_j \leq 1 \; \forall \; (i, j) \in E, \quad 0 \leq x_j \leq 1 \; (\forall \; j \in V), \quad \sum_{j \in V}(x_j) \; = \; k.$

Every vertex is a bin. Amount in bin $j = x_j$.

Function value for bin $j = g(x_j)$.

Minimise $Z = \sum_{j \in V} \; g(x_j)$.

$X = (x_1, x_2, \cdots, x_n)$.

(Needed) If for solution X, the function g achieves its minimum, then X should be an integer solution.

Helpful if g is a convex function; because a local min. is also a global min. for convex functions.

The sum of 2 convex functions is also a convex function.

Polynomial functions

$$g_1(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + C + \frac{b_1}{x} + \frac{b_2}{x^2} + \frac{b_3}{x^3} + \frac{b_4}{x^4}.$$

$$g_2(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + C + b_2\sqrt{x} + b_3\sqrt[3]{x} + b_4\sqrt[4]{x}.$$

$$(0 \le x \le 1) \longrightarrow (0 < w \le x \le 1).$$

Polynomial functions

$$g_1(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + C + \frac{b_1}{x} + \frac{b_2}{x^2} + \frac{b_3}{x^3} + \frac{b_4}{x^4}.$$

$$g_2(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + C + b_2\sqrt{x} + b_3\sqrt[3]{x} + b_4\sqrt[4]{x}.$$

$$(0 \leq x \leq 1) \longrightarrow (0 < w \leq x \leq 1).$$

Convexity:

Set the second derivative condition $g_1''(x) > 0$ as a constraint (or $g_2''(x) > 0$, whichever you use).

- for several values of x in the (0, 1) interval.

Outcome:

For g_1, obtained convex functions for small values of n, but not for $n = 150$.

For g_2, obtained convex functions with $n = 256$ (but depends on w).

Two stage process

Two stage process (to find integer optimal solution):

(a) Pick a value for w. Find coefficients a_i, b_j and C that satisfy certain conditions in LP form.

(If it doen't work, try a different value of w.)

Two stage process (to find integer optimal solution):

(a) Pick a value for w. Find coefficients a_i, b_j and C that satisfy certain conditions in LP form.

(If it doen't work, try a different value of w.)

(b) Use the coefficients to solve a (continuous) minimisation problem:

$$\begin{array}{c} \text{Minimise } Z = \sum_{j \in V} \; g(x_j) \\ x_i + x_j \;\leq\; (1+w) \;\; \forall\, (i,j) \in E \\ \forall\, j \in V,\; 0 < w \leq x_j \;\leq\; 1 \\ \sum_{j \in V} \; x_j \;=\; k + (n-k)w. \end{array} \tag{4}$$

(binary search on k)

Condition 1 to determine coefficients a_i, b_j and C

(1) Consider a bin (a vertex) i with a unit sized item ($x_i = 1$) and another bin with $x_j = w$. (Total $= 1 + w$)

Rearranging $(1 + w)$ into $[(0.9) + (w + 0.1)]$ should be more expensive.

$$g(1) + g(w) < g(0.9) + g(w + 0.1).$$

Condition 1 to determine coefficients a_i, b_j and C

(1) Consider a bin (a vertex) i with a unit sized item ($x_i = 1$) and another bin with $x_j = w$. (Total $= 1 + w$)

Rearranging $(1 + w)$ into $[(0.9) + (w + 0.1)]$ should be more expensive.

$g(1) + g(w) < g(0.9) + g(w + 0.1)$.

$g(1) + g(w) + eps \leq g(0.9) + g(w + 0.1)$.

Condition 1 to determine coefficients a_i, b_j and C

(1) Consider a bin (a vertex) i with a unit sized item ($x_i = 1$) and another bin with $x_j = w$. (Total $= 1 + w$)

Rearranging $(1 + w)$ into $[(0.9) + (w + 0.1)]$ should be more expensive.

$$g(1) + g(w) < g(0.9) + g(w + 0.1).$$

$$g(1) + g(w) + eps \leq g(0.9) + g(w + 0.1).$$

$$g(0.9) + g(w + 0.1) - g(1) - g(w) \geq eps.$$

Condition 1 to determine coefficients a_i, b_j and C

(1) Consider a bin (a vertex) i with a unit sized item ($x_i = 1$) and another bin with $x_j = w$. (Total $= 1 + w$)

Rearranging $(1 + w)$ into $[(0.9) + (w + 0.1)]$ should be more expensive.

$g(1) + g(w) < g(0.9) + g(w + 0.1)$.

$g(1) + g(w) + eps \leq g(0.9) + g(w + 0.1)$.

$g(0.9) + g(w + 0.1) - g(1) - g(w) \geq eps$.

$g(0.95) + g(w + 0.05) - g(1) - g(w) \geq eps$.

$g(0.85) + g(w + 0.15) - g(1) - g(w) \geq eps$.

$g(0.75) + g(w + 0.25) - g(1) - g(w) \geq eps$.

$g(0.65) + g(w + 0.35) - g(1) - g(w) \geq eps$.

$g(0.55) + g(w + 0.45) - g(1) - g(w) \geq eps$.

$(w \leq x_i \leq 1)$

Condition 1 to determine coefficients a_i, b_j and C

(actually we did it slightly differently)

$w_1 = 1 + w$.

Convex combination of $w_1 = p(1) + q(w)$, $p + q = 1$ and $p, q \geq 0$.

$g(0.9w_1) + g(0.1w_1) - g(1) - g(w) \geq eps$.

Condition 1 to determine coefficients a_i, b_j and C

(actually we did it slightly differently)

$w_1 = 1 + w$.

Convex combination of $w_1 = p(1) + q(w)$, $p + q = 1$ and $p, q \geq 0$.

$g(0.9w_1) + g(0.1w_1) - g(1) - g(w) \geq eps$.

$g(0.95w_1) + g(0.05w_1) - g(1) - g(w) \geq eps$.

$g(0.85w_1) + g(0.15w_1) - g(1) - g(w) \geq eps$.

$g(0.8w_1) + g(0.2w_1) - g(1) - g(w) \geq eps$.

$g(0.7w_1) + g(0.3w_1) - g(1) - g(w) \geq eps$.

$g(0.6w_1) + g(0.4w_1) - g(1) - g(w) \geq eps$.

$2 * g(0.5w_1) - g(1) - g(w) \geq eps$.

$(w \leq x_i \leq 1)$

Condition 2 to determine coefficients a_i, b_j and C

$w \leq$ item size in a bin ≤ 1.

Example: Consider one unit-sized bin and 5 w-sized bins.

Total volume $= 1 + 5w$.

Condition 2 to determine coefficients a_i, b_j and C

$w \leq$ item size in a bin ≤ 1.

Example: Consider one unit-sized bin and 5 w-sized bins.

Total volume $= 1 + 5w$.

Distribute this into 6 items (to be placed in 6 bins) of size $(1 + 5w)/6$ each.

$6g[(1+5w)/6] - g(1) - 5g(w) \geq eps$.

Condition 2 to determine coefficients a_i, b_j and C

$w \leq$ item size in a bin ≤ 1.

Example: Consider one unit-sized bin and 5 w-sized bins.

Total volume $= 1 + 5w$.

Distribute this into 6 items (to be placed in 6 bins) of size $(1 + 5w)/6$ each.

$6g[(1+5w)/6] - g(1) - 5g(w) \geq eps.$

$5g[(1+4w)/5] - g(1) - 4g(w) \geq eps.$

$4g[(1+3w)/4] - g(1) - 3g(w) \geq eps.$

$10g[(1+9w)/10] - g(1) - 9g(w) \geq eps.$

$20g[(1+19w)/20] - g(1) - 19g(w) \geq eps.$

$40g[(1+39w)/40] - g(1) - 39g(w) \geq eps.$

Condition 3 to determine coefficients a_i, b_j and C

Desired solution:

The unit-sized items occupy k bins. Total volume $= (1)(k) = k$.

The w-sized items occupy the remaining $(n-k)$ bins. Total volume $= (w)(n-k) = nw - kw$.

Condition 3 to determine coefficients a_i, b_j and C

Desired solution:

The unit-sized items occupy k bins. Total volume = $(1)(k) = k$.

The w-sized items occupy the remaining $(n-k)$ bins. Total volume = $(w)(n-k) = nw - kw$.

Condition 3 example:

Break every unit-sized item into p (=5) equal pieces of size $1/p$ (=0.2) each.

So now these 0.2 sized items occupy more bins ($5k$, **not** k).

The remaining volume of $nw - kw$ should be squeezed into fewer bins, that is, $(n-5k)$ bins, NOT $(n-k)$ any more.

Condition 3 to determine coefficients a_i, b_j and C

Desired solution:

The unit-sized items occupy k bins. Total volume $= (1)(k) = k$.

The w-sized items occupy the remaining $(n-k)$ bins. Total volume $= (w)(n-k) = nw - kw$.

Condition 3 example:

Break every unit-sized item into p (=5) equal pieces of size $1/p$ (=0.2) each.

So now these 0.2 sized items occupy more bins ($5k$, **not** k).

The remaining volume of $nw - kw$ should be squeezed into fewer bins, that is, $(n-5k)$ bins, NOT $(n-k)$ any more.

Fewer bins – so size in each bin will increase from w to $z_5 = \dfrac{nw-kw}{n-5k}$.

$5k * g(0.2) + (n-5k) * g(z_5) - k * g(1) - (n-k) * g(w) \geq eps.$

Similarly, for various values of p, as long as $w \leq x_i \leq 1$.

LP to determine coefficients

Combining all conditions, we get an LP with the coefficients a_1, a_2, a_3, a_4, C, b_1, b_2, b_3 and b_4 as the unknowns.

For a given (n, k) combination, we experiment with different values of w and eps – pick the (w, eps) for which we get a feasible solution to the LP.

LP to determine coefficients

Combining all conditions, we get an LP with the coefficients a_1, a_2, a_3, a_4, C, b_1, b_2, b_3 and b_4 as the unknowns.

For a given (n, k) combination, we experiment with different values of w and eps – pick the (w, eps) for which we get a feasible solution to the LP.

Does this process — finding coefficients for a given (n, k) — need to run in polynomial time?

Perhaps not; once we have the coefficients for a given (n, k), in principle, we can re-use them for numerous instances (in step 2) for which n and k are the same.

LP to determine coefficients

Combining all conditions, we get an LP with the coefficients a_1, a_2, a_3, a_4, C, b_1, b_2, b_3 and b_4 as the unknowns.

For a given (n, k) combination, we experiment with different values of w and eps – pick the (w, eps) for which we get a feasible solution to the LP.

Does this process — finding coefficients for a given (n, k) — need to run in polynomial time?

Perhaps not; once we have the coefficients for a given (n, k), in principle, we can re-use them for numerous instances (in step 2) for which n and k are the same.

Software used:

Step 1 (Linear Program): GLPK first, Gurobi later.

Step 2 (Non-linear Minimisation): MINOS.

Step 2: Minimise a non-linear function to get an integer solution

Desired solution value $= (k)g(1) + (n-k)g(w)$. (known)

Step 2: Minimise a non-linear function to get an integer solution

Desired solution value $= (k)g(1) + (n-k)g(w)$. (known)

$$\begin{aligned} \text{Minimise } Z = \textstyle\sum_{j \in V} \; g(x_j) & \\ x_i + x_j \;\leq\; (1+w) \;\; \forall\, (i,j) \in E & \\ \textstyle\sum_{j \in V} \; x_j \;=\; k + (n-k)w & \\ \textstyle\sum_{j \in V} \; g(x_j) \;=\; (k)g(1) + (n-k)g(w) & \\ \forall\, j \in V, \; 0 < w \leq x_j \;\leq\; 1. & \end{aligned} \tag{5}$$

MINOS optimisation is sensitive to the initial solution.

We used $x_i = 1$ (for every i) as the initial solution.

What if we are unable to find an integer solution for k?

- During binary search, what if we are unable to find an integer solution for k?

  Can we conclude that the optimal solution value is $< k$?

  No, unfortunately (not yet) :-(

What if we are unable to find an integer solution for k?

- During binary search, what if we are unable to find an integer solution for k?

  Can we conclude that the optimal solution value is $< k$?

  No, unfortunately (not yet) :-(

- **Not** necessary to obtain a $(\mathbf{w}, \mathbf{1})$ solution.

  For an edge (3, 7), $x_3 + x_7 \leq (1 + w)$.

  If $x_3 > 0.5(1 + w)$, which means that $x_7 < 0.5(1 + w)$
  $\longrightarrow$ include vertex 3 in the independent set, but **not** 7.

  True for every neighbour of vertex 3 in the graph.

Instances Tested

- A 25-vertex instance: Convex function in Step 1, and integer optimal solutions in Step 2.

Instances Tested

- A 25-vertex instance: Convex function in Step 1, and integer optimal solutions in Step 2.
- A 150-vertex instance with opt 29: No convex function in Step 1;

  For $k = 20$, $w = 0.008$, obtained integer solution. (using function g_1)

  For $k = 29$, obtained non-integer solution which can be easily converted to an integer solution. (using function g_2 and an initial solution for 5/150 vertices).

Instances Tested

- A 25-vertex instance: Convex function in Step 1, and integer optimal solutions in Step 2.
- A 150-vertex instance with opt 29: No convex function in Step 1;

  For $k = 20$, $w = 0.008$, obtained integer solution. (using function g_1)

  For $k = 29$, obtained non-integer solution which can be easily converted to an integer solution. (using function g_2 and an initial solution for 5/150 vertices).
- A 53-vertex instance with opt 26:

  For $10 \leq k \leq 26$, we obtained integer solution with the parameters from the 150-vertex instance (but with slightly different $w = 0.19$).

Instances Tested

- A 25-vertex instance: Convex function in Step 1, and integer optimal solutions in Step 2.

- A 150-vertex instance with opt 29: No convex function in Step 1;

  For $k = 20$, $w = 0.008$, obtained integer solution. (using function g_1)

  For $k = 29$, obtained non-integer solution which can be easily converted to an integer solution. (using function g_2 and an initial solution for $5/150$ vertices).

- A 53-vertex instance with opt 26:

  For $10 \leq k \leq 26$, we obtained integer solution with the parameters from the 150-vertex instance (but with slightly different $w = 0.19$).

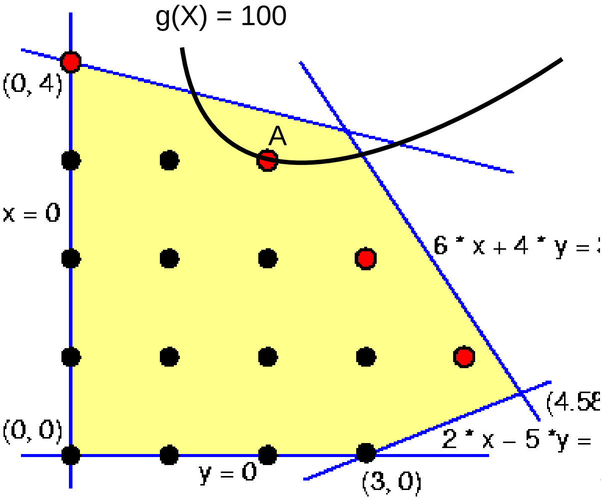
- A 256-vertex instance with opt 30:

  For ($k \leq 22$): Able to re-use the parameters (AND the corresponding function g_1) from the 150-vertex instance.

  ($k = 30$): Testing continues.

LP polytope: Generate functions to get integer solution?

Point A is the integer solution that we desire. If A is in the interior, then NO matter what function g you use, there are an infinite number of feasible solutions with $g(X) = 100$! (A is NOT unique)

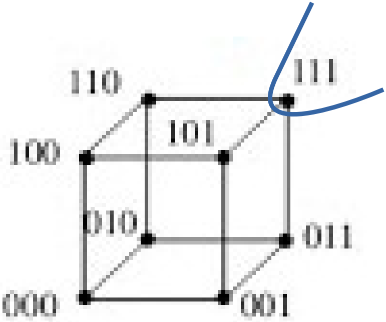


Generating functions for integer solutions - 2

But luckily, $x_i^* = 1$ for at least one $i \in V$. $(x_i \leq 1.)$

Every feasible integer solution X is a vertex of the LP polytope. $X = (x_1, \cdots, x_n) =$ (zeroes and ones).

So there is hope that we can design a function whose intersection with the LP polytope is just a "small" region around X^*.



$[g(X) = \text{constant value}]$

3-SAT (Satisfiability) Model

Recently, we started modelling the M.I.S. as a Satisfiability problem (3-SAT, in particular).

You can follow the results on Researchgate:

https://www.researchgate.net/publication/380034972
(the 3-SAT approach)

https://www.researchgate.net/publication/361555319
(the continuous non-linear optimisation approach)

Thank you for listening!