Formulating the minimum weight feedback vertex set on undirected graphs as an integer linear programming problem

Alexandru Luchianov

Mentor: Professor Emil Kelevedjiev

SRS, Bulgaria, Aprilci
25 July - 14 August 2021

1 Introduction

Linear programming is a method to achieve the best outcome under a given set of constraints when the outcome function and all the constraints are linear inequalities. Usual linear programming is solvable in polynomial time. However, when the variables can be only integers or only binary variables the problem becomes NP-complete [1].

The feedback vertex set of a graph is a set of vertices of minimal size such that each cycle passes through at least one of these vertices. The problem is NP-complete, but it is APX-complete [2] in the undirected case. We will now present an integer programming formulation of the undirected case with a polynomial number of constraints. The minimum weight feedback vertex set is a feedback vertex set such that the sum of the weights of the nodes in the set is minimal.

2 Existing formulations

The existing formulations are similar to the formulations already used to transform the minimum spanning tree and the Steiner tree into integer linear programming problems. However, more efficient formulations for both of these problems already exist.[3]

2.1 Cycle elimination formulation

Let w_v for each $v \in V$ be the weight of node v. We define x_v for each $v \in V$ as whether the node v is eliminated or not, then for each cycle we add the condition that at least one node has to be eliminated. The formal model is:

$$\begin{aligned} &\text{minimize} && \sum_{v \in V} x_v * w_v \\ &\text{subject to} && \sum_{v \in C} x_v \geq 1, && \forall C \text{ ,where C is a cycle in graph} \\ & && x_v \text{ is a binary variable, } \forall v \in V \end{aligned}$$

This formulation is short and concise, however it uses an exponential number of constraints.

2.2 Subset bound formulation

We define x_v the same as before. For each edge i between a_i and b_i we create 2 variables c_i and d_i. We call an edge "active" if both endpoints are not eliminated. c_i is 1 if and only if the edge is active and d_i is an auxiliary variable. For each subset of nodes we add the condition that the number of active edges inside must be smaller than the number of nodes.

$$
\begin{array}{lll}
\text{minimize} & \sum_{v \in V} x_v * w_v & \\
\text{subject to} & 2c_i + d_i + x_a + x_b = 2 & i = 1 \ldots m \\
& \sum_{\substack{1 \le i \le m \\ a_i, b_i \in S}} c_i < |S| & \forall \text{ subset } S \text{ of nodes} \\
& c_v \text{ is a binary variable, } 1 \le i \le m & \\
& d_v \text{ is a binary variable, } 1 \le i \le m & \\
& x_v \text{ is a binary variable, } \forall v \in V &
\end{array}
$$

The subset bound formulation is also exponential in number of constraints, but it has the advantage that we don't have to perform any special preprocessing on the graph beforehand (such as finding all cycles).

3 Proposed Formulation

As all the existing formulations are exponential in number of constraints, we will now present a formulation that uses $\mathcal{O}(N + M)$ constraints.

The main issue with creating a polynomial model for this problem is finding the appropriate conditions for a set of nodes to be a feedback vertex set. To find a convenient condition we will define $ndel$ as the number of deleted nodes, $mleft$ as the number of active edges and $cleft$ as the number of connected components after deleting the set of nodes.

A set of nodes is a feedback vertex set if and only if $ndel + mleft + cleft = n$. If instead of deleting the nodes in the set, we only delete the edges adjacent to them, the condition becomes $mleft + cleft = n$. It is easy to see that $mleft + cleft \le n$.

We can count the number of connected components by using a modified flow model. We will define the following variables:

- f_i = how much flow goes from a_i to b_i
- f_{m+i} = how much flow goes from b_i to a_i
- fin_v = how much flow goes from the source to the node v
- $fout_v = 1$ how much flow goes from the node v to the sink
- $in_v = \begin{cases} 1 & \text{if } 0 < fin_v \\ 0 & \text{if } otherwise \end{cases}$
- $cflow = \sum_{v \in V} in_v$

We will also add the condition that how much flow enters node v should be equal to how much flow leaves node v.

It is easy to see that by formulating the flow as above $cflow$ can take any value in the interval $[cleft, n]$. Because $mleft + cleft \leq n$ we can rewrite the condition $mleft + cleft = n$ as $mleft + cflow = n$.

The mathematical model is written as follows:

$$
\begin{array}{lll}
\text{minimize} & \sum_{v \in V} x_v * w_v & \\
\text{subject to} & 2c_i + d_i + x_a + x_b = 2 & i = 1 \ldots m \\
& fin_v - fout_v + \sum_{\substack{1 \leq i \leq 2m \\ b_i = v}} f_i - \sum_{\substack{1 \leq i \leq 2m \\ a_i = v}} f_i = 0 & \forall v \in V \\
& cflow - \sum_{v \in V} in_v = 0 & \\
& cflow + \sum_{1 \leq i \leq m} c_i = n & \\
& c_v \text{ is a binary variable, } 1 \leq i \leq m & \\
& d_v \text{ is a binary variable, } 1 \leq i \leq m & \\
& x_v \text{ is a binary variable, } \forall v \in V & \\
& in_v \text{ is a binary variable, } \forall v \in V & \\
& f_i \in \{0, n\}^{2m} & \\
& fin \in \{0, n\}^n &
\end{array}
$$

4 Benchmarking

For all of the above formulations you can find programs that convert an instance of the feedback vertex set problem into an integer linear programming model at `https://github.com/alexluchianov/FVS-ILP`.

Execution times in miliseconds				
n	m	Cycle base model	Subset-based model	Proposed model
2	1	6.20 ms	5.60 ms	5.90 ms
3	3	5.50 ms	5.20 ms	6.80 ms
4	6	6.20 ms	6.20 ms	9.40 ms
5	8	5.80 ms	8.20 ms	12.40 ms
6	15	7.30 ms	11.20 ms	55.30 ms
7	16	8.10 ms	15.30 ms	66.30 ms
8	28	11.60 ms	84.80 ms	720.30 ms
9	28	11.50 ms	136.10 ms	619.20 ms
10	28	16.10 ms	424.90 ms	1092.90 ms
11	28	19.90 ms	1017.70 ms	1195.20 ms
12	20	9.70 ms	986.60 ms	1585.00 ms
12	30	21.00 ms	2524.60 ms	1921.40 ms
12	66	384.60 ms	16347.50 ms	22014.10 ms
13	57	834.40 ms	37846.70 ms	14766.60 ms
14	57	1983.00 ms	170132.50 ms	20558.50 ms
14	80	4245.50 ms	318383.60 ms	76127.20 ms
15	63	7672.60 ms	- ms	43359.20 ms
15	86	15413.20 ms	- ms	146127.00 ms
16	63	21622.30 ms	- ms	104012.60 ms
17	73	93919.00 ms	- ms	178118.00 ms
17	100	614752.00 ms	- ms	1120441.00 ms
18	77	296848.00 ms	- ms	185570.00 ms

The table was generated by testing the models on 10 random graphs for $n \leq 16$ and by testing only one random graph for $n \geq 17$. The $-$ sign means that the subset-based model was so slow that the actual execution time did not matter.

The models were tested using the integer linear programming solver available at `lpsolve.sourceforge.net/5.5/`.

At first the exponential models outperform the proposed model because the heuristics used by the solver perform better on simpler models, however as the number of nodes increases the proposed model begins to overtake the other models.

References

[1] Karp, Richard M. (1972)"Reducibility Among Combinatorial Problems" Proc. Symposium on Complexity of Computer Computations, IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., New York: Plenum, pp. 85–103

[2] Bafna, Vineet; Berman, Piotr; Fujito, Toshihiro (1999), "A 2-approximation algorithm for the undirected feedback vertex set problem", SIAM Journal on Discrete Mathematic

[3] Tamer F. Abdelmaguid , "Solving Integer Linear Programs by Exploiting Variable-Constraint Interactions: A Survey"