Finding violated odd-subset constraints

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

March 2017
Odd subset constraints

The MAXCUT problem can be formulated using variables

\[ x_e = \begin{cases} 
1 & \text{if } e \text{ is in the cut} \\
0 & \text{otherwise} 
\end{cases} \]

One class of constraints for this problem is the set of odd subset constraints:

\[ \sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e \leq |F| - 1 \quad (1) \]

where \( C \) is a chordless cycle in the graph and \( F \subseteq C \) with \( |F| \) odd. This document describes the Barahona-Mahjoub [1] routine for finding violated constraints of this form.
Finding a violated constraint: duplicate the graph

A solution $x$ for the current relaxation of the maxcut problem is known. These are regarded as the edge weights.

The routine duplicates the original graph, along with these weights. Thus, each edge $i$ now has a copy $i'$. If $(i,j)$ is an edge in the original graph, then edges are introduced between vertices $i$ and $j'$ and between $i'$ and $j$ with weights $1 - x_{ij}$. 
An example
For example, consider the graph with $x_e$ shown:
Duplicating the example graph

This graph is duplicated and connecting edges are added. In the picture, we indicate just three of the edges between the original graph and its copy. The number of such edges is actually twice the number of edges in the original graph.

\[ \sum x_e - \sum x_e \leq |F| - 1 \]

Want these small: small weighted black edge.

Want these large: small weighted blue edge.

F: edges going between original graph & duplicate.

Path from \( V_j \) to \( V'_j \): cycle.
Finding a violated constraint: look for short paths

For each vertex $i$, the algorithm looks for a shortest path between vertex $i$ and vertex $i'$.

Any such path of length smaller than one can be converted into a violated constraint of the form (1).

In particular, the edges of the path are the edges of the cycle in the original graph.

The subset $F$ consists of the edges in the path with one endpoint in the original nodes and one endpoint in the duplicated set of nodes.
A short path between $v_1$ and $v_1'$

This graph contains a path of length 0.7 from vertex $v_1$ to vertex $v_1'$, as shown in the next picture. Since the path length is smaller than 1.0, we have a violated constraint.
Finding a violated constraint

The cycle $C$ is thus $\{v_1, v_6, v_2, v_7, v_3, v_4, v_1\}$ and the edges in the set $F$ are $(v_1, v_6), (v_2, v_6)$ and $(v_1, v_4)$. The size of $F$ is three, but the current value of

$$\sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e$$

is $0.9 + 0.8 + 1 - 0.1 - 0.1 - 0.2 = 2.3 > |F| - 1$.

Notice that the violation of the constraint is 0.3. This is exactly the shortfall between the path length 0.7 and the threshold value 1.0. Stated differently, the violation of the constraint and the path length add up to one.
Exercise

Show that the sum of the violation of the constraint and the path length is always equal to one.

It follows that if there are any violated constraints then the corresponding path will have length smaller than one.

Thus, searching for the shortest path will lead to the violated constraint.
Heuristics

The Barahona-Mahjoub routine is expensive, so heuristics are often used to find violated constraints. (See [2], for example.)

One heuristic is a breadth-first search: starting at a vertex, grow a tree through the graph using only edges with $x_e$ close to zero or one.

If the branches of the tree meet, a cycle has been formed.

If there is an odd number of edges on this cycle with $x_e$ close to one then we have a candidate for violation of (1).
Large xe: candidates for F
Small xe: candidates for CNF
Finding a cycle in the example with the heuristic

Say in the graph above, we put an edge in the tree if $x_e \leq 0.2$ or $x_e \geq 0.8$. Then this will give edges around the outside of the graph, together with the edge $(v_3, v_5)$.

In particular, starting from $v_1$ with an empty tree, we first add edges $(v_1, v_4)$ and $(v_1, v_6)$ to the tree.

Growing from $v_4$ adds the edge $(v_4, v_3)$.

Growing from $v_6$ adds the edge $(v_2, v_6)$.

Growing from $v_3$ adds the edges $(v_3, v_5)$ and $(v_3, v_7)$.  

The tree so far

$v_3 \rightarrow v_7$ with weight 0.1
$v_3 \rightarrow v_4$ with weight 0.1
$v_4 \rightarrow v_5$ with weight 0.1
$v_4 \rightarrow v_1$ with weight 1
$v_1 \rightarrow v_6$ with weight 0.9
$v_1 \rightarrow v_2$ with weight 0.8
Extending the tree, getting a cycle
Now growing the tree from vertex $v_2$ leads to the addition of edge $(v_2, v_7)$, which creates the cycle we saw before.
Now we have our odd cutset

Now growing the tree from vertex $v_2$ leads to the addition of edge $(v_2, v_7)$, which creates the cycle we saw before.
The heuristic is a heuristic

This heuristic is a lot quicker than the Barahona-Mahjoub algorithm.

However, it may miss a constraint.

For example, if we change the thresholds from 0.2 and 0.8 to 0.1 and 0.9 then we would not have found the cycle;

if we make the thresholds too large, then the tree becomes too dense and there is an excess of choice, which makes it hard to identify the violated constraint.

F. Barahona and A. R. Mahjoub.  
On the cut polytope.  

J. E. Mitchell.  
Computational experience with an interior point cutting plane algorithm.  