Cluster Editing

Heuristic Solution

Algorithm 1 Main solver

Input: G, t, w

$G \leftarrow \texttt{kernelize}(G)$
for each connected component C_i **do**
    $S_i \leftarrow \texttt{trivial\_solution}(C_i)$
while timeout is not reached **do**
    for each connected component C_i **do**
        $j \leftarrow \texttt{sample\_weights}(w_1, \ldots, w_\ell)$
        $S' \leftarrow \texttt{bfs\_greedy}(C_i, S_i, t[j])$
        if $\texttt{cost}(S') < \texttt{cost}(S_i)$ **then**
            $S_i \leftarrow S'$
            $w[j] \leftarrow w[j] + 1$
return $\mathcal{S} = \bigcup_i S_i$

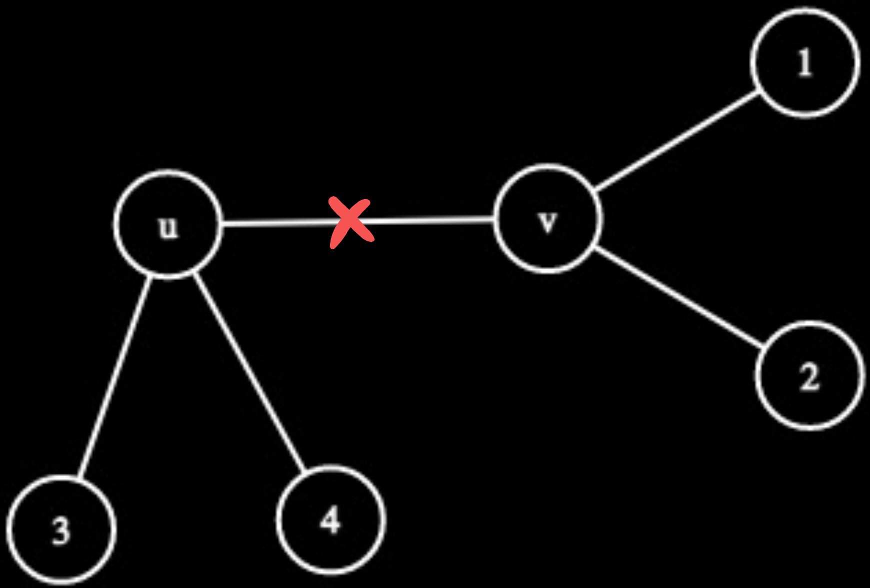
Algorithm 2 `bfs_greedy` heuristic

Input: C, S, t

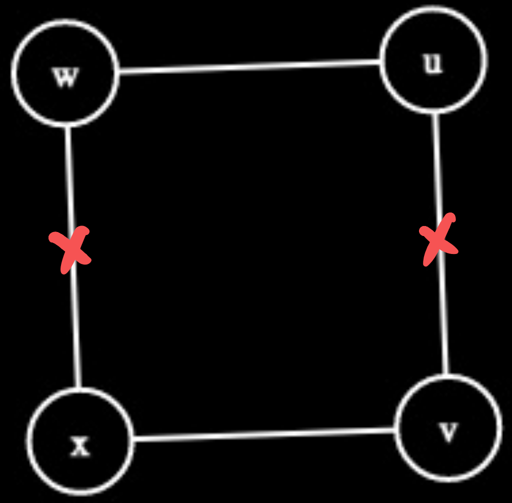
$seen \leftarrow [false] * n$
for each $v \in C$ in a random order **do**
    if $seen[v]$ **then**
        go to the next vertex
    $X \leftarrow \texttt{select\_BFS}(C, v, seen, t)$
    $S' \leftarrow S$
    for u in X **do**
        $S' \leftarrow \texttt{isolate}(u, S')$
    for u in X **do**
        $S' \leftarrow \texttt{best\_move}(u, S')$
return S'

```
self.cost_calculate = (self.n * (self.n - 1)) / 2 - self.graph.m()
```

1. Let u be a vertex with either a 1-neighbor or two adjacent 2-neighbors. If u has another neighbor v such that u and v have no common neighbor, then delete uv.

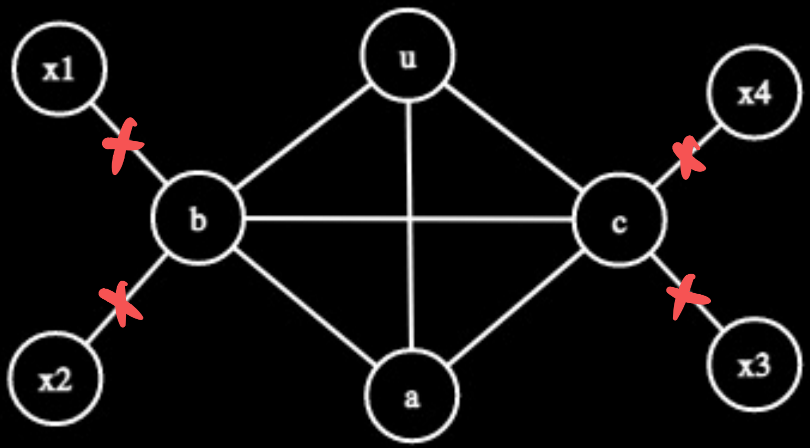


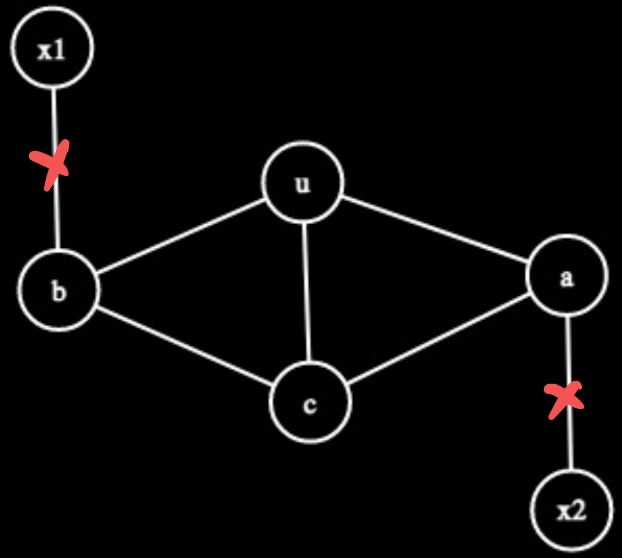
2. Let uvxw be an induced C4 where v and w have degree 2. Delete uv and wx.



3. Let u be a 3-vertex with neighbors a, b, c.

- If ab, bc and ac are edges, a has degree 3 and b, c both have degree at most 5, delete all edges bx and cx for x not in (u, a, b, c).
- If ac and bc are edges, ab is a non-edge, and a, b, c all have degree at most 3, delete all edges ax and bx for x not in (u, c).





4. Let K be a clique on k vertices such that each vertex outside of K has at most one neighbor in K. For all of u in K, denote by f(u) the number of neighbours of u outside of k. $K = u_i, ..., u_k \text{where} f(u_1) \leqslant ... \leqslant f(u_k)$. For every i in $[1, k]$, $\sum_{j=i+1}^{k} f(u_i) \leqslant \binom{k}{2} - \binom{i}{2}$ delete all edges with exactly one endpoint in K