Bron Kerbosch Algorithm (BK)

The BK algorithm [1] uses the recursive backtracking paradigm to enumerate all maximal cliques in the graph. At any given point in time it maintains three lists, $C$, $I$, and $X$. The set $C$ contains the vertices of the clique currently being enumerated, the set $I$ contains vertices that are connected to all vertices in $C$ and can be added to $C$ to make a larger clique and the set $X$ contains vertices that are connected to all vertices in $C$ but are excluded from being added to $C$ because all cliques containing vertices in $X$ have already been enumerated in a different recursion cycle. Algorithm 1 gives an overview of the algorithm.

Algorithm 1: Bron and Kerbosch Algorithm

1 Algorithm: BK Algorithm
   \[\text{Input: A unweighted undirected graph } G = (V, E)\]
   \[\text{Output: A list of all maximal cliques in } G\]
2 $C = \emptyset$; /* A set of vertices that represent a maximal clique or can be extended to a maximal clique */
3 $I = V(G)$; /* The set of vertices that are connected to all vertices in $C$ and can be added to $C$ to make a larger clique */
4 $X = \emptyset$; /* The set of vertices connected to all vertices in $C$ but excluded from being added to $C */
5 $BK$ - Enumerate($C, I, X$)

Algorithm 2: The recursive function utilized in the BK algorithm

1 Algorithm: $BK$ - Enumerate($C, I, X$)
2 if $I = \emptyset$ and $X = \emptyset$ then
3     print $C$ as maximal clique;
4 else
5     $v$ = vertex connected to maximum number of vertices in $I$;
6     while $v \neq \emptyset$ do
7         $BK$ - Enumerate($C \cup \{v\}, I \cap N(v), X \cap N(v)$);
8         /* $N(v)$ represents the neighbors of vertex $v$ in $G$ */
9     $I = P - \{v\}$;
10    $X = N \cup \{v\}$;
11    $v$ = vertex connected to maximum number of vertices in $I$

The current recursion stops when $C$ cannot be expanded any further, i.e., $I$ becomes $\emptyset$. At this point, if set $X$ is also $\emptyset$ then the vertices $C$ form a maximal clique and added to the output set. The condition $X = \emptyset$ checks for maximality.
because if \( X \) were not empty then it would mean the \( C \) can be further expanded with the vertices from \( X \) to form an even larger clique and hence \( C \) cannot be maximal.

**References**