

Enumeration Algorithms for Maximal Matching in Chain Graphs

Kanata Maruyama Naoto Kaneko Hiroshi Fujiwara Katsuhisa Ouchi

Uno provided an algorithm that for a bipartite graph $G = (V, E)$ with $V = L \cup R$, enumerates all maximal matchings in $O(|V|)$ time per matching [1]. A *chain* graph is a bipartite graph $G = (V, E)$ with $V = L \cup R$ such that, for each of the vertex sets L and R , the neighborhoods of any two vertices satisfy an inclusion relation. In this study, we propose an algorithm, Algorithm 1, that efficiently enumerates all maximal matchings of a chain graph.

Algorithm 1 EnumMaximalMatching(G, M)

```

1: Input : a chain graph  $G = (V, E)$  with  $V = L \cup R$ , and an edge set  $M$ 
2: if  $G$  has no edges then
3:   output  $M$  and return
4: end if
5: choose a vertex  $v$  of maximum degree from the one with the fewer number of vertices from  $L$  and  $R$ 
6: for each edge  $e$  incident to  $v$  do
7:    $M' \leftarrow M \cup \{e\}$ 
8:   obtain  $G'$  from  $G$  by removing the endpoints of  $e$  and all their incident edges, and then deleting
     any isolated vertices
9:   ENUMMAXIMALMATCHING( $G', M'$ )
10: end for

```

Theorem 1 *The algorithm ENUMMAXIMALMATCHING enumerates all maximal matchings in the input chain graph in $O(1)$ time per matching.*

To achieve this running time, we employ a data structure that efficiently manages the degree structure of the chain graph. As illustrated in Fig. 1, the vertex sets L and R are maintained as doubly linked lists consisting of vertex nodes and boundary nodes. A vertex node represents a single vertex in V , while a boundary node represents a contiguous block of vertices with the same degree. To perform adjacency tests between pairs of vertices in constant time, we employ an adjacency matrix.

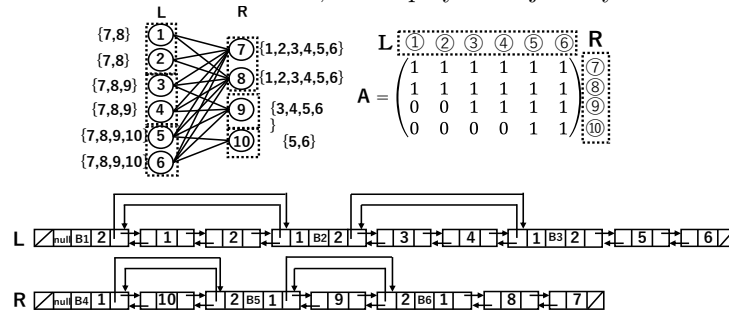


Figure 1: The data structure used by the algorithm ENUMMAXIMALMATCHING to store the degree sequence of a chain graph.

References

[1] T. Uno. Algorithms for Enumerating All Perfect, Maximum and Maximal Matchings in Bipartite Graphs. In *Proc. ISAAC '97*, LNCS 1350, pp. 92–101, Springer, 1997.