Enumeration of Minimal Tree Decompositions of Complete Graphs

A.J. Petrenjuk

Higher College of Civil Aviation
Kirovograd, Ukraine

We describe an algorithm of constructive enumeration of minimal tree decompositions of K_{2n}. This algorithm was implemented by a FORTRAN program which enabled us to obtain a full list of nonisomorphic minimal tree decompositions of K_6. The list consists of 19 decompositions.

1. We use the term 'graph' in the sense of Harary [1].

Let $\mathbf{G} = \{G_1, \ldots, G_s\}$ be a family of graphs, and let H be a graph with a nonempty edge set. The subgraphs $H_1, \ldots, H_m$ of H form a $\mathbf{G}$-decomposition of H if (1) every H_i is isomorphic to a member of $\mathbf{G}$, (2) H_i, H_j have no edge in common ($1 \leq i < j \leq m$), and (3) $\bigcup H_i = H$. The graphs $H_1, \ldots, H_m$ are called *components*, and the number m is called the *size* of the $\mathbf{G}$-decomposition. When H admits a $\mathbf{G}$-decomposition, we denote by $g(H, \mathbf{G})$ the minimal size of such a decomposition. A $\mathbf{G}$-decomposition of H whose size is $g(H, \mathbf{G})$ is called *minimal*.

2. We are interested in the list $L(H, \mathbf{G})$ of all nonisomorphic minimal $\mathbf{G}$-decompositions of H, and in the number $N(H, \mathbf{G}) = |L(H, \mathbf{G})|$. Here we consider the special problem of the constructive enumeration of minimal $\mathbf{T}$-decompositions of K_v where $\mathbf{T}$ is the set of trees. Put $V(K_v) = \{1, 2, \ldots, v\}$.

Some constructions of such decompositions (with isomorphic components) may be found in [3].

Beineke [2] proved that $g(K_v, \mathbf{T}) = \lceil v/2 \rceil$ for all $v \geq 1$. Obviously the components of the minimal $\mathbf{T}$-decompositions of K_{2n} are the spanning trees only. It is easy to prove that if G is a component of a minimal $\mathbf{T}$-decomposition of K_{2n} then $\Delta(G) < n$ where $\Delta(G)$ is the maximum vertex degree of G.

As a consequence, of the 6 nonisomorphic trees on 6 vertices (cf. [1]), only 4 can be components of minimal $\mathbf{T}$-decompositions of K_6.

3. Let us now formulate our algorithm which permits us to construct $L(K_{2n}, \mathbf{T})$. Let $Q_1, \ldots, Q_r$ be the list of all trees which may appear as components of minimal $\mathbf{T}$-decompositions of K_{2n}. We assume that there is an algorithm that generates, without repetition, all the permutations of the set $V(K_{2n})$ in some fixed order

$$\phi_0, \phi_1, \ldots, \phi_l$$

where $l = (2n)! - 1$ and ϕ_0 is the identity.

Before formulating the algorithm, let us introduce a linear order $\ll$ in the set of pairs (Q_i, ϕ_j) so that $(Q_i, \phi_j) \ll (Q_p, \phi_r)$ if and only if $i < p$ or $i = p$,

$j < r$. We call a *trace* the linear arrangement of pairs (Q, ϕ) in accordance with the order.

0. Put $t = 1$, $S = \emptyset$, $m = 1$. Construct the triple of objects $(t, Q^{(t)}, \phi^{(t)}) = (1, Q_1, \phi_0)$.
1. $t \leftarrow t + 1$; go to 2.
2. If $t = 1$ go to 11; if $1 < t \leq n$ go to 3; otherwise go to 9.
3. If the pair $(Q^{(t-1)}, \phi^{(t-1)})$ is the last along the trace, go to 7.; otherwise go to 4.
4. Let (Q, ϕ) directly follow $(Q^{(t-1)}, \phi^{(t-1)})$ along the trace.
5. If the tree Q_ϕ has no edge in common with each of the trees $Q^{(1)}\phi^{(1)}, \ldots, Q^{(t-1)}\phi^{(t-1)}$, put $(t, Q^{(t)}, \phi(t)) = (t, Q, \phi)$ and go to 1.; otherwise go to 6.
6. If (Q, ϕ) is not the last pair along the trace, denote by (Q, ϕ) the next pair and go to 5.; otherwise go to 7.
7. If $t > 1$ do $t \leftarrow t - 1$ and go to 8.; otherwise go to 11.
8. Denote $(Q, \phi) = (Q^{(t)}, \phi^{(t)})$ and
9. Construct the canonical form R (see section 4 below) of the minimal $\mathbf{T}$-decomposition $Q^{(1)}\phi^{(1)}, \ldots, Q^{(n)}\phi^{(n)}$ and go to 10.
10. If $R \in S$ go to 7.; otherwise add R to S and go to 7.
11. Stop. The result is $S = L(K_{2n}, \mathbf{T})$.

Remark: One may shorten the work of the algorithm by deleting from the trace in advance all pairs (Q, ϕ) with equal Q_ϕ's except for one.

4. Let us represent the edges of a graph in the form ij where $i < j$. Then a tree can be written as a lexicographically ordered sequence of edges. The lexicographic order induces naturally an order in the set of spanning trees of K_{2n}.

A $\mathbf{T}$-decomposition of K_{2n} is called *reduced* if its components are written in the lexicographic order. In a natural way, a lexicographic order is induced in the set of reduced minimal $\mathbf{T}$-decompositions of K_{2n}.

The smallest one among all the reduced minimal $\mathbf{T}$-decompositions which are isomorphic to D, is called the *canonical form* of D.

The canonical form is an invariant in the set of minimal $\mathbf{T}$-decompositions of K_{2n}. It is a complete invariant, i.e. it distinguishes the latter completely.

5. The algorithm described in section 3 was implemented as a FORTRAN program. The canonical form was constructed by the subroutine KANON. The main program and KANON both use the subroutine PERM (due to V.Bol'shakov [4]) which generates the permutations of order v.

The computation was made for the case $2n = 6$ resulting in the following.

Theorem. $N(K_6, \mathbf{T}) = 19$.

6. Below we present the list $L(K_6, \mathbf{T})$. The third components of the decompositions are not listed since they can be easily determined. The last column gives

the order of the automorphism group of the corresponding decomposition.

1.	12 23 34 45 56	13 14 25 26 36	1
2.	12 23 34 45 56	13 14 25 35 46	1
3.	12 23 34 45 56	13 16 24 25 35	1
4.	12 23 34 45 56	13 25 26 35 46	1
5.	12 23 34 45 56	13 24 25 35 46	2
6.	12 23 34 45 56	13 15 25 26 46	6
7.	12 23 34 45 56	14 26 35 36 46	1
8.	12 23 34 45 56	16 24 25 35 36	1
9.	12 23 34 45 56	13 14 15 24 26	1
10.	12 23 34 45 56	16 24 35 36 46	1
11.	12 23 34 45 56	16 24 26 35 36	1
12.	12 23 25 45 56	13 26 35 36 46	6
13.	12 23 25 45 56	14 15 16 26 34	1
14.	12 23 25 45 56	13 14 26 35 36	1
15.	13 23 34 45 56	12 24 25 36 46	1
16.	13 23 34 45 56	12 14 25 26 35	1
17.	13 23 34 45 56	12 15 16 35 46	1
18.	12 23 34 36 45	13 14 15 25 46	3
19.	12 23 34 45 56	13 16 24 35 46	6

References

1. F. Harary, "Graph Theory", Addison-Wesley, 1969.
2. L.W. Beineke, *Decomposition of complete graphs into forests*, Magyar Tud. Akad. Mat. Kut. Int. Közl **9** (1964), 589–594.
3. C. Huang and A. Rosa, *Decomposition of complete graphs into trees*, Ars Combinat. **5** (1978), 23–63.
4. V.I. Bol'shakov, "Kombinatornye vycisleniya na EVM, Glava 1, Paket prikladnyh programm po kombinatorike", MGU, Moskva, 1985. (scientific report in Russian).