ALGORITHMS FOR GENERALIZED STABILITY NUMBERS OF TREE GRAPHS

D. E. DAYKIN AND C. P. NG

(Received 2 April 1965)

1. Introduction

In this paper we give some algorithms for determining $\alpha^w(T)$ and $\beta^w(T)$, the generalized internal and external stability numbers respectively, of a finite directed tree graph T whose nodes are weighted by a function w. We define $\alpha^w(T)$ and $\beta^w(T)$ in section 2. When w gives every node of T the weight 1 then $\alpha^w(T) = \alpha(T)$ and $\beta^w(T) = \beta(T)$ where $\alpha(T)$ and $\beta(T)$ are the usual stability numbers.

K. Maghout has given [2] an exceedingly elegant technique for evaluating $\alpha(G)$ and $\beta(G)$ for any finite graph G. However in applications his method requires a considerable number of Boolean operations. It was this fact which led us to look for new ways of evaluating $\alpha(T)$, $\beta(T)$ and we feel that our results are exceedingly simple to apply to tree graphs. Also if a graph G can be reduced to a family of trees by the removal of a small number of nodes, then our methods evaluate $\alpha^w(G)$ and $\beta^w(G)$ more simply than K. Maghout's evaluates $\alpha(G)$ and $\beta(G)$.

2. Definitions

This section is devoted to explaining the notation used in our algorithms. Once this notation is understood one can use the algorithms without reading their proofs.

The letters G and T will stand for a general graph and a tree graph respectively. Both will be finite and directed, and any node of T will be taken as its root ρ. A subset S of G is said to be internally stable if no two nodes of S are adjacent. We will write $\mathscr{I}(G)$ for the family of all internally stable subsets of G. The internal stability number (or the coefficient of internal stability [1], or the independence number [3]) of G is the maximum number of elements of an internally stable set, i.e.,

$$(1) \qquad \alpha(G) = \max \{|S|;\ S \in \mathscr{I}(G)\}.$$

We say that a graph G is weighted if with each node $\nu \in G$, there is associated a non-negative real number $w(\nu)$, and we write

$$w(S) = \sum_{\nu \in S} w(\nu) \text{ for } S \subseteqq G.$$

Then we define the generalized internal stability number $\alpha^w(G)$ of G to be the maximum of the weights of the internally stable sets of G, i.e.,

(2) $$\alpha^w(G) = \max \{w(S); S \in \mathscr{I}(G)\}.$$

We write $(\nu, \mu) \in G$ to mean that ν, μ are nodes of G joined by an edge directed from ν to μ. This edge is directed both ways if $(\nu, \mu), (\mu, \nu) \in G$. A subset S of G is said to be externally stable if for every node $\nu \notin S$ there is a node $\mu \in S$ such that $(\nu, \mu) \in G$. We will write $\mathscr{E}(G)$ for the family of all externally stable subsets of G. The external stability number (or the coefficient of external stability [1], or the domination number [3]) of G is the minimum number of elements of an externally stable set, i.e.,

(3) $$\beta(G) = \min \{|S|; S \in \mathscr{E}(G)\}.$$

We define the generalized external stability number $\beta^w(G)$ of G to be the minimum of the weights of the externally stable sets of G, i.e.,

(4) $$\beta^w(G) = \min \{w(S); S \in \mathscr{E}(G)\}.$$

There may be several sets S such that $w(S) = \alpha^w(G)$, however any such set will be maximal, i.e. $S \subseteqq R$, $S \neq R$ implies $R \notin \mathscr{I}(G)$. Maghout's method produces simultaneously all maximal sets of $\mathscr{I}(G)$, and once these are known one can clearly determine $\alpha^w(G)$. Similar remarks apply to $\beta^w(G)$. Clearly if $w(\nu) = 1$ for all $\nu \in G$ then $\alpha^w(G) = \alpha(G)$ and $\beta^w(G) = \beta(G)$.

For each $\nu \in T$, we write $d(\nu)$ for the (undirected) distance of ν from the root ρ. Also we let $B(\nu)$ denote the subgraph of T which is that *branch* of T with ν as root, i.e., the set $B(\nu)$ consists of all μ of T such that $d(\mu)-d(\nu) \geqq 0$, and that any undirected path joining μ to ρ passes through ν. Clearly $B(\nu)$ is itself a tree. We define $A(\nu)$ to be the set of nodes μ of T *immediately above* ν, i.e., such that $d(\mu) = d(\nu)+1$ and μ is adjacent to ν. Further we let $A\!\uparrow\!(\nu)$ denote the set of nodes μ of $A(\nu)$ such that $(\nu, \mu) \in T$ (upwards) or $(\nu, \mu), (\mu, \nu) \in T$ (symmetric). In symmetric trees $A\!\uparrow\!(\nu) = A(\nu)$. We denote by ν' the node *immediately below* ν, i.e., such that $d(\nu') = d(\nu)-1$ and ν' is adjacent to ν.

Finally, we specify a tree as follows. We number the nodes in any way, $1, 2, \cdots, n = |T|$, and then prepare a table by writing the numbers, $1, 2, \cdots, n$ on the top row, and $A(i)$ under the number i. This specification of a tree is different from the one given in [4]. It does not indicate the directing of the tree and will only be used in section **3**.

Our work splits naturally into two parts dealing with internal and external stability respectively.

PART I. INTERNAL STABILITY

3. Algorithms for $\alpha(T)$

Using the specification of a tree given above one can find a set $S \in \mathscr{I}(T)$ such that $\alpha(T) = |S|$ by

ALGORITHM 1. *Find an unprimed number i in the top row such that $A(i) = \phi$. Then give this number i a prime. Further for each j, such that $i \in A(j)$ delete j from every set $A(k)$ in which it occurs, and delete j from the top row, and delete $A(j)$ from the table. Repeat the operation as many times as possible. If S is the set of primed numbers remaining in the top row then $|S| = \alpha(T)$ and $S \in \mathscr{I}(T)$.*

We do not prove this algorithm as it is merely a reformulation of

ALGORITHM 2. *Given T, initially put $R = S = \phi$. Then* (i) *adjoin nodes of valency* 1 *(except the root ρ if it is also of valency* 1) *of $T-R$ to S, or adjoin ρ to S if $T-R$ consists of ρ only, and* (ii) *adjoin to R the nodes of S and those nodes of T adjacent to S. Repeat the operation* (i), (ii) *until $R = T$ when $|S| = \alpha(T)$ and $S \in \mathscr{I}(T)$.*

Notice that in algorithm 1 the number i with $A(i) = \phi$ specifies a node of valency 0 or 1 and we adjoin the node to S by giving i a prime. This is equivalent to operation (i) in algorithm 2. Also we remove the nodes i and the nodes j adjacent to i by deleting the numbers $A(j)$ and j from the table in algorithm 1. This is equivalent to operation (ii) in algorithm 2.

PROOF OF ALGORITHM 2. If two nodes of T are adjacent, and one of them is adjoined to S at some stage of the construction, then if the other is not already in R it is immediately adjoined to R. Hence both of them cannot be in S. This proves that S is internally stable.

Suppose now we have reached some stage of the construction of S. Let ν be a node of $T-R$ of valency 0 or 1. In the latter case let μ be the node of $T-R$ adjacent to ν. We cannot now do better than adjoin ν to S, for at most one of ν, μ can be in S, and if we adjoin μ to S this may prevent not only ν but other nodes of $T-R$ from being adjoined to S. Since this is true at each stage of the construction the result follows.

4. The generalized internal stability number

In this section we discuss some properties of $\alpha^w(G)$. Our first result concerns sums and products of weighting functions.

THEOREM 1. *If v, w are weighting functions defined on a graph G, then*

(i) $$\alpha^{v+w}(G) \leqq \alpha^{v}(G)+\alpha^{w}(G),$$

and

(ii) $$\alpha^{vw}(G) \leqq \alpha^{v}(G)\alpha^{w}(G).$$

PROOF. By definition (2), there are sets $P, Q, R, S \in \mathscr{I}(G)$ such that

$$\alpha^{v}(G) = v(P); \quad \alpha^{w}(G) = w(Q);$$
$$\alpha^{v+w}(G) = (v+w)(R) \text{ and } \alpha^{vw}(G) = vw(S),$$

where, for example,

$$v(P) = \sum_{\nu \in P} v(\nu) \text{ and } (v+w)(R) = \sum_{\nu \in R} [v(\nu)+w(\nu)].$$

Hence

$$\alpha^{v+w}(G) = v(R)+w(R) \leqq v(P)+w(Q) = \alpha^{v}(G)+\alpha^{w}(G),$$

and

$$\alpha^{vw}(G) = \sum_{\nu \in S} [v(\nu)w(\nu)] \leqq v(S)w(S) \leqq v(P)w(Q)$$
$$= \alpha^{v}(G)\alpha^{w}(G).$$

A simple result which is very useful in practice is given in

THEOREM 2. *If C is a complete subgraph of G with $|C| = r+1$, and if ν is a node of C of valency r in G such that $w(\nu) \geqq w(\mu)$ for all $\mu \in C$, then*

$$\alpha^{w}(G) = w(\nu)+\alpha^{w}(G-C).$$

PROOF. If $S \in \mathscr{I}(G)$ then, since C is complete, at most one node of C is in S. Moreover if $S \cap C = \phi$ then $S+\nu \in \mathscr{I}(G)$. Hence

$$\alpha^{w}(G) = \max\{w(S); S \in \mathscr{I}(G), S \cap C \neq \phi\}$$
$$= w(\nu)+\max\{w(S); S \in \mathscr{I}(G-C)\}$$
$$= w(\nu)+\alpha^{w}(G-C),$$

and the theorem is proved.

It should be noted that we have restricted our attention to non-negative weighting function w. The reason for this is that if there was a node ν such that $w(\nu) \leqq 0$ then we would get the following simplification:

$$\alpha^{w}(G) = \max\{w(S); S \in \mathscr{I}(G)\} = \max\{w(S); S \in \mathscr{I}(G), \nu \notin S\}$$
$$= \max\{w(S); S \in \mathscr{I}(G-\nu)\} = \alpha^{w}(G-\nu).$$

Similar remarks apply to $\beta^{w}(G)$.

5. Algorithms for tree graphs

We first give the algorithm for evaluating $\alpha^{w}(T)$, for a given tree T with weighting function w, as

ALGORITHM 3. *For each $\nu \in T$, starting with terminal nodes and working progressively towards the root, define a pair $(m(\nu), n(\nu))$ of real numbers by the recursive relations*

$$m(\nu) = w(\nu) + \sum_{\mu \in A(\nu)} n(\mu), \tag{5}$$

and

$$n(\nu) = \sum_{\mu \in A(\nu)} \max \{m(\mu), n(\mu)\}. \tag{6}$$

Then

$$\alpha^w(T) = \max \{m(\rho), n(\rho)\}. \tag{7}$$

When ν is a terminal node, then $A(\nu) = \phi$ and the sums in (5), (6) are zero.

Once the pairs $(m(\nu), n(\nu))$ have all been defined, we can immediately find an internally stable set S for which $\alpha^w(T) = w(S)$. The idea of the method is to decide which nodes are to be in S, by working node by node up the tree from the root. The method is given as

ALGORITHM 4. *A set $S \in \mathscr{I}(T)$ with $\alpha^w(T) = w(S)$ can be selected recursively as follows. If $\mu = \rho$ or if $\mu \in A(\nu)$ and $\nu \notin S$, then we decide whether $\mu \in S$ or $\mu \notin S$ by means of relations* (8) *below. If $\nu \in S$, then $\mu \notin S$ for all $\mu \in A(\nu)$.*

$$\left\{ \begin{array}{l} \textit{If } m(\mu) > n(\mu) \textit{ then } \mu \in S; \\ \textit{if } m(\mu) < n(\mu) \textit{ then } \mu \notin S; \\ \textit{if } m(\mu) = n(\mu) \textit{ then } \mu \in S \textit{ or } \mu \notin S \textit{ arbitrarily.} \end{array} \right. \tag{8}$$

One can obtain all sets S with $\alpha^w(T) = w(S)$ by taking both of the choices, $\mu \in S$ and $\mu \notin S$, separately in (8) whenever $m(\mu) = n(\mu)$. If $m(\nu) \neq n(\nu)$ for all $\nu \in T$ then S is unique.

PROOF OF ALGORITHM 3. We simply show that the pairs of numbers $(m(\nu), n(\nu))$ constructed by the algorithm have the property

$$\left. \begin{array}{rl} m(\nu) & = \max \{w(S); S \in \mathscr{I}(B(\nu)), \nu \in S\} \\ \text{and } n(\nu) & = \max \{w(S); S \in \mathscr{I}(B(\nu)), \nu \notin S\} \end{array} \right\} \text{ for } \nu \in T. \tag{9}$$

Trivially (9) holds for terminal nodes ν. To prove recursively that (9) holds for all $\nu \in T$, we now suppose that τ is a non-terminal node and that (9) holds for all nodes $\nu \in B(\tau)$, $\nu \neq \tau$. Now by construction (5) we have

$$\begin{aligned} m(\tau) &= w(\tau) + \sum_{\mu \in A(\tau)} n(\mu) \\ &= w(\tau) + \sum_{\mu \in A(\tau)} \max \{w(S); S \in \mathscr{I}(B(\mu)), \mu \notin S\}, \quad \text{by (9)} \\ &= w(\tau) + \max \sum_{\mu \in A(\tau)} \{w(S); S \in \mathscr{I}(B(\mu)), \mu \notin S\} \\ &= w(\tau) + \max \{w(S); S \in \mathscr{I}(B(\tau)), A(\tau) \cap S = \phi, \tau \notin S\} \\ &= \max \{w(S); S \in \mathscr{I}(B(\tau)), \tau \in S\}, \end{aligned}$$

and
$$\begin{aligned} n(\tau) &= \sum_{\mu\in A(\tau)} \max\{m(\mu), n(\mu)\} \\ &= \sum_{\mu\in A(\tau)} \alpha^w(B(\mu)) \\ &= \sum_{\mu\in A(\tau)} \max\{w(S); S\in \mathscr{I}(B(\mu))\} \\ &= \max\{w(S);\ S\in\mathscr{I}(B(\tau)), \tau\notin S\}, \end{aligned}$$

and the algorithm is established.

PROOF OF ALGORITHM 4. First we obtain expressions for $\alpha^w(T)$ in each of three cases.

Case 1. $m(\rho) > n(\rho)$. Here ρ is in every set S constructed by the algorithm, and, using in turn (7), (8), (5), (6) and (7), we have

$$\begin{aligned} \alpha^w(T) &= \max\{m(\rho), n(\rho)\} = m(\rho) = w(\rho) + \sum_{\mu\in A(\rho)} n(\mu) \\ &= w(\rho) + \sum_{\mu\in A(\rho)} \Big\{ \sum_{\tau\in A(\mu)} \max\{m(\tau), n(\tau)\}\Big\} \\ &= w(\rho) + \sum_{\mu\in A(\rho)} \Big\{ \sum_{\tau\in A(\mu)} \alpha^w(B(\tau))\Big\}, \qquad \rho\in S. \end{aligned} \tag{10}$$

Case 2. $m(\rho) < n(\rho)$. In this case ρ is not in any set S constructed by the algorithm. Moreover, using (7), (8), (6) and (7) in turn, we get

$$\begin{aligned} \alpha^w(T) &= \max\{m(\rho), n(\rho)\} = n(\rho) = \sum_{\mu\in A(\rho)} \max\{m(\mu), n(\mu)\} \\ &= \sum_{\mu\in A(\rho)} \alpha^w(B(\mu)), \qquad \rho\notin S. \end{aligned} \tag{11}$$

Case 3. $m(\rho) = n(\rho)$. We can choose at will to have ρ in or not in the set S we construct by the algorithm, and (10) or (11) holds accordingly.

Now if we let the height $h(T)$ of a tree T be given by

$$h(T) = \max\{d(\nu); \nu\in T\},$$

then the construction given in the algorithm is trivially valid for trees T with $h(T) = 0$. Suppose as an induction hypothesis that T has height $h > 0$, and that the method is valid for all trees of height $< h$. Then for each $\mu\in A(\rho)$ by using the method we can obtain a set S_μ such that $\alpha^w(B(\mu)) = w(S_\mu)$. Hence if (11) holds the union of sets $S_\mu\in\mathscr{I}(B(\mu))$ gives a choice of S. Similarly if (10) holds then for each $\mu\in A(\rho)$ and $\tau\in A(\mu)$ we can construct a set $S_{\mu\tau}\in\mathscr{I}(B(\tau))$ such that $\alpha^w(B(\tau)) = w(S_{\mu\tau})$, and the union of the sets $S_{\mu\tau}$ and ρ gives a choice of S. Since no node $\mu\in A(\rho)$ belongs to S when (10) holds, this establishes the algorithm.

6. Remarks on evaluating $\alpha^w(G)$

We can always remove from G a set R of nodes such that $G-R$ is a family of trees, and the following remarks are of value when $|R|$ is small. Now

$$\begin{aligned}\alpha^w(G) &= \max\{w(S);\, S\in\mathscr{I}(G)\}\\ &= \max_{Q\subseteq R,\, Q\in\mathscr{I}(G)}\{w(Q)+\max\{w(S);\, S\in\mathscr{I}(G-R-j(Q))\}\}\\ &= \max_{Q\subseteq R,\, Q\in\mathscr{I}(G)}\{w(Q)+\alpha^w(G-R-j(Q))\},\end{aligned}$$

where $j(Q)$ is the set of nodes of $G-R$ adjacent to Q in G. Clearly the sets $G-R-j(Q)$ are families of trees. If $|R|$ is small there will be few choices for Q. Hence in this case it may be convenient to evaluate $\alpha^w(G)$ by using our algorithms to evaluate $\alpha^w(G-R-j(Q))$ for each Q in the equation above.

PART 2. EXTERNAL STABILITY

7. An algorithm for $\beta(T)$

Given a tree graph T, to evaluate $\beta(T)$ we may use

ALGORITHM 5. *A set $S\in\mathscr{E}(T)$ such that $\beta(T)=|S|$ may be constructed by initially putting $R=S=\phi$, and then proceeding as follows,*

(i) *if ν is a terminal node of $T-R$ and μ is the node adjacent to ν in $T-R$ adjoin μ to S if $(\nu,\mu)\in T-R$, otherwise adjoin ν to S;*

(ii) *adjoin all pairs μ,ν to R for which $(\mu,\nu)\in T-R$ and $\nu\in S$;*

(iii) *if $\nu\in T-R$, and the valency of ν is 0 in $T-R$ adjoin ν to S.*

Repeat the operation (i), (ii), (iii) *until $R=T$ when $|S|=\beta(T)$ and $S\in\mathscr{E}(T)$.*

PROOF. Trivially S is externally stable by (ii), (iii). Now suppose that we are at some stage of the construction, and ν, μ are as defined in (i). Then at least one of ν,μ must be adjoined to S. If $(\nu,\mu)\in T-R$ the natural choice is μ, for then μ will *take care* of ν and possibly some other nodes of the tree. Since this is true at each stage of the construction the result follows.

The table specification of a tree is useful for the external stability of only symmetric trees, so we do not discuss it here.

8. Algorithms for $\beta^w(T)$

In this section we show how to evaluate $\beta^w(T)$ and to find a set $S\in\mathscr{E}(T)$ such that $w(S)=\beta^w(T)$. The method for finding $\beta^w(T)$ is given as

ALGORITHM 6. *For each node $\nu\in T$, starting with terminal nodes and* *working progressively towards the root ρ, define a triple $(p(\nu), q(\nu), r(\nu))$ of real numbers by the recursive relations,*

$$(12) \qquad p(\nu) = w(\nu) + \sum_{\mu \in A(\nu)} \min\{p(\mu), q(\mu), r(\mu)\},$$

$$\begin{matrix}(13)\\(14)\end{matrix} \qquad q(\nu) = \begin{cases} \infty & (\text{if } A\!\uparrow\!(\nu) = \phi), \\ p(\lambda) + \sum_{\substack{\mu \in A(\nu) \\ \mu \neq \lambda}} \min\{p(\mu), q(\mu)\} & (\text{otherwise}), \end{cases}$$

$$\begin{matrix}(15)\\(16)\end{matrix} \qquad r(\nu) = \begin{cases} \infty & (\text{if } (\nu, \nu') \notin T), \\ \sum_{\mu \in A(\nu)} \min\{p(\mu), q(\mu)\} & (\text{otherwise}), \end{cases}$$

where λ is a node of $A\!\uparrow\!(\nu)$ such that $p(\lambda) - q(\lambda) \leqq p(\mu) - q(\mu)$ for all $\mu \in A\!\uparrow\!(\nu)$. Then

$$(17) \qquad \beta^w(T) = \min\{p(\rho), q(\rho)\}.$$

Notice that $r(\rho) = \infty$ because $(\rho, \rho') \notin T$. When ν is a terminal node then $A(\nu) = A\!\uparrow\!(\nu) = \phi$ so that the sums in (12) and (16) are empty. Therefore $p(\nu) = w(\nu)$, $q(\nu) = \infty$, and $r(\nu)$ is 0 or ∞ for terminal nodes ν. Also when $(\nu, \nu') \in T$, $A\!\uparrow\!(\nu) \neq \phi$, and there is a $\lambda \in A\!\uparrow\!(\nu)$ such that $p(\lambda) \leqq q(\lambda)$ then $q(\nu) = r(\nu)$. We always have either $r(\nu) = \infty$ or $r(\nu) \leqq q(\nu)$.

Once the triples $(p(\mu), q(\mu), r(\mu))$ have been defined we can then find a set $S \in \mathscr{E}(T)$ such that $w(S) = \beta^w(T)$ by working up the tree from the root. The method corresponds to that of algorithm 4 and is given as

ALGORITHM 7. *One set $S \in \mathscr{E}(T)$ with $w(S) = \beta^w(T)$ is determined by the following recursive rules:*

(i) *First, $\rho \in S$ iff $p(\rho) \leqq q(\rho)$.*

(ii) *If $\nu \in S$ and $\mu \in A(\nu)$, then $\mu \in S$ iff $p(\mu) \leqq q(\mu), r(\mu)$.*

(iii) *If $\nu \notin S$, $r(\nu) < p(\nu), q(\nu)$ and $\mu \in A(\nu)$, then $\mu \in S$ iff $p(\mu) \leqq q(\mu)$.*

(iv) *If $\nu \notin S$ and $r(\nu) \nless p(\nu), q(\nu)$, choose a node $\lambda \in A\!\uparrow\!(\nu)$ such that $p(\lambda) - q(\lambda) \leqq p(\mu) - q(\mu)$ for all $\mu \in A\!\uparrow\!(\nu)$. Then $\lambda \in S$, whilst if $\mu \in A(\nu)$, $\mu \neq \lambda$, then $\mu \in S$ iff $p(\mu) \leqq q(\mu)$.*

PROOF OF ALGORITHM 6. We simply show that for each $\nu \in T$ the numbers $p(\nu)$, $q(\nu)$, $r(\nu)$ constructed by the algorithm have the properties,

$$(18) \qquad \begin{cases} p(\nu) = \min\{w(S); \nu \in S, S \in \mathscr{E}(B(\nu))\}, \\ q(\nu) = \min\{w(S); \nu \notin S, S \in \mathscr{E}(B(\nu))\}, \\ r(\nu) = \min\{w(S); \nu \notin S, S + \nu' \in \mathscr{E}(B(\nu) + \nu'), (\nu, \nu') \in T\}. \end{cases}$$

Thus, corresponding to $m(\nu)$ and $n(\nu)$ in (9), the numbers $p(\nu)$ and $q(\nu)$ are the minimum of the weights of the externally stable sets of the branch $B(\nu)$ which do and do not respectively contain ν. The number $r(\nu)$ is similar

https://doi.org/10.1017/S1446788700004031 Published online by Cambridge University Press

in nature to $q(\nu)$, except that it anticipates that when we consider the branch $B(\nu')$ we will be able to let ν' *take care of* ν.

Trivially (18) hold for terminal nodes ν. To prove recursively that (18) hold for all $\nu \in T$, we suppose that τ is a non-terminal node and that (18) hold for all nodes $\nu \in B(\tau)$, $\nu \neq \tau$. Then we have

$$(19) \quad \min\{p(\mu), q(\mu)\} = \min\{w(S); S \in \mathscr{E}(B(\mu))\} \text{ for } \mu \in B(\tau), \mu \neq \tau.$$

Using our induction hypothesis on (18), and its consequence (19), in definition (12) we obtain

$$(20) \quad \begin{aligned} p(\nu) = w(\nu) + \sum_{\mu \in A(\nu)} \min\,[&\min\{w(S); S \in \mathscr{E}(B(\mu))\}, \\ &\min\{w(S); \mu \notin S, S+\nu \in \mathscr{E}(B(\mu)+\nu), (\mu, \nu) \in T\}]. \end{aligned}$$

Let μ be any one of the nodes of $A(\nu)$. If $S \in \mathscr{E}(B(\mu))$ then $S+\nu \in \mathscr{E}(B(\mu)+\nu)$ and so

$$\min\{w(S); S \in \mathscr{E}(B(\mu))\} \geqq \min\{w(S); S+\nu \in \mathscr{E}(B(\mu)+\nu)\}.$$

On the other hand, if $S+\nu \in \mathscr{E}(B(\mu)+\nu)$ but $S \notin \mathscr{E}(B(\mu))$ then ν must have been ***taking care*** of μ, and so $\mu \notin S$, $(\mu, \nu) \in T$. It follows from (20) by this argument that

$$\begin{aligned} p(\nu) &= w(\nu) + \sum_{\mu \in A(\nu)} \min\{w(S); S+\nu \in \mathscr{E}(B(\mu)+\nu)\} \\ &= \min\{w(S); S \in \mathscr{E}(B(\nu)), \nu \in S\}. \end{aligned}$$

If $A\!\nmid(\nu) = \phi$ then there is no set $S \in \mathscr{E}(B(\nu))$ with $\nu \notin S$. We indicate the impossibility of obtaining such a set by putting $q(\nu) = \infty$ in (13). If $A\!\nmid(\nu) \neq \phi$, then

$$\begin{aligned} q(\nu) &= p(\lambda) + \sum_{\substack{\mu \in A(\nu) \\ \mu \neq \lambda}} \min\{p(\mu), q(\mu)\}, && \text{by (14)}, \\ &= \min_{\eta \in A\nmid(\nu)} [p(\eta) + \sum_{\substack{\mu \in A(\nu) \\ \mu \neq \eta}} \min\{p(\mu), q(\mu)\}], && \text{by definition of } \lambda, \\ &= \min_{\eta \in A\nmid(\nu)} [\min\{w(S); S \in \mathscr{E}(B(\eta)), \eta \in S\} \\ &\qquad + \sum_{\substack{\mu \in A(\nu) \\ \mu \neq \eta}} \min\{w(S); S \in \mathscr{E}(B(\mu))\}], && \text{by (18), (19)}, \\ &= \min_{\eta \in A\nmid(\nu)} [\min\{w(S); S \in \mathscr{E}(B(\nu)), \nu \notin S, \eta \in S\}] \\ &= \min\{w(S); S \in \mathscr{E}(B(\nu)), \nu \notin S\}. \end{aligned}$$

Finally we show that $r(\nu)$ also satisfies (18). When $(\nu, \nu') \notin T$ we indicate the impossibility of obtaining a set S of the type required in (18) by putting $r(\nu) = \infty$ in (15). On the other hand, if $(\nu, \nu') \in T$, then

https://doi.org/10.1017/S1446788700004031 Published online by Cambridge University Press

$$
\begin{aligned}
r(\nu) &= \sum_{\mu \in A(\nu)} \min \{p(\mu), q(\mu)\}, && \text{by (16)},\\
&= \sum_{\mu \in A(\nu)} \min \{w(S); S \in \mathscr{E}(B(\mu))\}, && \text{by (19)},\\
&= \min \{w(S); S \in \mathscr{E}(B(\nu)-\nu)\}\\
&= \min \{w(S); S+\nu' \in \mathscr{E}(B(\nu)+\nu'), \nu \notin S\}, && \text{since } (\nu, \nu') \in T,
\end{aligned}
$$

and this completes the proof of algorithm 6.

PROOF OF ALGORITHM 7. Trivially the algorithm is valid for trees of height 0. Suppose as an induction hypothesis that T has height h and that the algorithm is valid for all trees of height $< h$. Also let S be the subset of T constructed by the rules (i)—(iv) of the algorithm. In each of the two cases below, we show that S has the desired properties, namely, that $S \in \mathscr{E}(T)$ and $w(S) = \beta^w(T)$.

Case 1. $q(\rho) < p(\rho)$. Here $\rho \notin S$ by rule (i), and since $r(\rho) = \infty$ the nodes μ of $A(\rho)$ belong, or do not belong, to S according to rule (iv). Since $q(\rho) < p(\rho)$ we have $q(\rho) \neq \infty$ and so (13) shows that $A \nmid (\rho) \neq \phi$. Hence the λ described in rule (iv) does exist and thus λ is in S by rule (iv).

For $\mu \in A(\rho)$, $\mu \neq \lambda$, by rule (iv) we have $\mu \in S$ iff $p(\mu) \leqq q(u)$, and $r(\mu)$ is not needed. The rules (i)—(iv) will then determine a subset S_μ of $B(\mu)$, and since μ is the root of $B(\mu)$, by our induction hypothesis $S_\mu \in \mathscr{E}$ $(B(\mu))$ and $w(S_\mu) = \beta^w(B(\mu))$.

If $p(\lambda) > q(\lambda)$ and we follow the rules (i)—(iv) treating $B(\lambda)$ as a tree with λ as root we will obtain a subset of $B(\lambda)$ which does not contain λ. This fact would prevent us from using our induction hypothesis. We overcome the difficulty as follows. Whatever the relative values of $p(\lambda)$ and $q(\lambda)$, and solely for the purpose of proving the algorithm, we let $B^*(\lambda)$ denote the tree obtained from $B(\lambda)$ by changing the weight of the node λ to 0. Also we will use a star * to denote the effect of changing from $B(\lambda)$ to $B^*(\lambda)$. Then $w^*(\mu) = w(\mu)$, $p^*(\mu) = p(\mu)$, $q^*(\mu) = q(\mu)$, $r^*(\mu) = r(\mu)$ for all $\mu \in B(\lambda)$ except that $w^*(\lambda) = 0$, $p^*(\lambda) = p(\lambda) - w(\lambda)$. Moreover comparision of (12) and (13) shows that $p^*(\lambda) \leqq q^*(\lambda)$ since $w^*(\lambda) = 0$. The rules (i)—(iv) enable us to construct a set S^*_λ of $B^*(\lambda)$ with $\lambda \in S^*_\lambda$. By our induction hypothesis $S^*_\lambda \in \mathscr{E}(B^*(\lambda))$ and $w^*(S^*_\lambda) = \beta^{w^*}(B^*(\lambda))$. Now by inspection of rules (i)—(iv) it can be seen that S^*_λ is precisely that subset S_λ of S which is contained in $B(\lambda)$. Since $\mathscr{E}(B^*(\lambda)) = \mathscr{E}(B(\lambda))$ we have $S_\lambda \in \mathscr{E}(B(\lambda))$. Moreover since $\lambda \in S_\lambda$ it follows that $w(S_\lambda) = w(\lambda) + w^*(S_\lambda)$.

Now let $S = \bigcup_{\mu \in A(\rho)} S_\mu$. For all $\mu \in A(\rho)$ we have $S_\mu \in \mathscr{E}(B(\mu))$, and so if $\eta \in B(\mu)$ then either $\eta \in S_\mu \subseteqq S$ or there is a $\tau \in S_\mu$ such that $(\eta, \tau) \in T$. Also $(\rho, \lambda) \in T$, and so $S \in \mathscr{E}(T)$.

Now

$$
\begin{aligned}
\beta^w(T) &= q(\rho), && \text{by (17)},\\
&= p(\lambda) + \sum_{\substack{\mu\in A(\rho)\\ \mu\neq\lambda}} \min\{p(\mu), q(\mu)\}, && \text{by (14)},\\
&= w(\lambda) + p^*(\lambda) + \sum_{\substack{\mu\in A(\rho)\\ \mu\neq\lambda}} \beta^w(B(\mu)), && \text{by definition of } p^*(\lambda) \text{ and (17)},\\
&= w(\lambda) + \beta^{w^*}(B^*(\lambda)) + \sum_{\substack{\mu\in A(\rho)\\ \mu\neq\lambda}} w(S_\mu)\\
&= w(\lambda) + w^*(S_\lambda) + \sum_{\substack{\mu\in A(\rho)\\ \mu\neq\lambda}} w(S_\mu)\\
&= w(S_\lambda) + \sum_{\substack{\mu\in A(\rho)\\ \mu\neq\lambda}} w(S_\mu)\\
&= \sum_{\mu\in A(\rho)} w(S_\mu) = w(S),
\end{aligned}
$$

and so we have shown that $S\in\mathscr{E}(T)$ and $w(S) = \beta^w(T)$ as required.

Case 2. $p(\rho) \leqq q(\rho)$. Here $\rho\in S$ by rule (i), and the nodes μ of $A(\rho)$ belong, or do not belong, to S according to rule (ii). It will be convenient to let (i) P, (ii) Q, and (iii) R be the sets of nodes $\mu\in A(\rho)$ with (i) $p(\mu)\leqq q(\mu)$, $r(\mu)$, (ii) $q(\mu) < p(\mu)$, $q(\mu)\leqq r(\mu)$, and (iii) $r(\mu) < p(\mu)$, $q(\mu)$ respectively. Notice that P, Q, R are disjoint sets with union $A(\rho)$, for if $p(\mu)\not\leqq q(\mu)$, $r(\mu)$, we have either $q(\mu)\leqq r(\mu)$ or $r(\mu) < q(\mu)$.

For $\mu\in P$, we have $\mu\in S$ by rule (ii). The rules (i)—(iv) will then construct a subset S_μ of $B(\mu)$ with $\mu\in S_\mu$, and since μ is the root of $B(\mu)$, by induction hypothesis $S_\mu\in\mathscr{E}(B(\mu))$ and $w(S_\mu) = \beta^w(B(\mu))$.

For $\mu\in Q$, we have $\mu\notin S$ by rule (ii). As already proved in Case 1 the rules (i)—(iv) allow us to construct a subset S_μ of $B(\mu)$ such that $\mu\notin S_\mu$, $S_\mu\in\mathscr{E}(B(\mu))$ and $w(S_\mu) = \beta^w(B(\mu))$.

Finally, for $\eta\in R$, we have $\eta\notin S$ by rule (ii). Since $r(\eta) < p(\mu)$, $q(\mu)$ definitions (15), (16) show that $(\eta,\rho)\in T$. Now let $\tau\in A(\eta)$. We have $\tau\in S$ iff $p(\tau)\leqq q(\tau)$ and $r(\tau)$ is not needed. Again the rules (i)—(iv) will construct a subset $S_{\eta\tau}$ of $B(\tau)$. By our induction hypothesis, since τ is the root of $B(\tau)$ we have $S_{\eta\tau}\in\mathscr{E}(B(\tau))$ and $w(S_{\eta\tau}) = \beta^w(B(\tau))$.

Now let

$$S = \{\rho\} \cup \{\bigcup_{\mu\in P,Q} S_\mu\} \cup \{\bigcup_{\eta\in R}\{\bigcup_{\tau\in A(\eta)} S_{\eta\tau}\}\}.$$

For all $\mu\in P, Q$ and $\eta\in R$, $\tau\in A(\eta)$, we have $S_\mu\in\mathscr{E}(B(\mu))$ and $S_{\eta\tau}\in\mathscr{E}(B(\tau))$. Hence to show that $S\in\mathscr{E}(T)$ we need only consider the nodes $\eta\in R$. For each $\eta\in R$, since $(\eta,\rho)\in T$ and $\rho\in S$ the node η is *taken care* of by ρ and this shows that $S\in\mathscr{E}(T)$.

In conclusion

$$
\begin{aligned}
\beta^w(T) &= p(\rho), \text{ by (17)},\\
&= w(\rho)+\sum_{\mu\in A(\rho)} \min\{p(\mu), q(\mu), r(\mu)\}, \qquad \text{by (12)},\\
&= w(\rho)+\sum_{\mu\in P} p(\mu)+\sum_{\mu\in Q} q(\mu)+\sum_{\eta\in R} r(\eta), \text{ by definition of } P, Q, R,\\
&= w(\rho)+\sum_{\mu\in P,Q} \min\{p(\mu), q(\mu)\}+\sum_{\eta\in R}\sum_{\tau\in A(\eta)} \min\{p(\tau), q(\tau)\}\\
&= w(\rho)+\sum_{\mu\in P,Q} \beta^w(B(\mu))+\sum_{\eta\in R}\sum_{\tau\in A(\eta)} \beta^w(B(\tau))\\
&= w(\rho)+\sum_{\mu\in P,Q} w(S_\mu)+\sum_{\eta\in R}\sum_{\tau\in A(\eta)} w(S_{\eta\tau})\\
&= w(\{\rho\})+w(\{\bigcup_{\mu\in P,Q} S_\mu\})+w(\{\bigcup_{\eta\in R}\{\bigcup_{\tau\in A(\eta)} S_{\eta\tau}\}\})\\
&= w(S),
\end{aligned}
$$

and the proof is complete.

References

[1] C. Berge, *The Theory of Graphs* (London, 1962).

[2] K. Maghout, "Sur la détermination des nombres de stabilité et du nombre chromatique d'un graphe" *C. R. Acad. Sci. Paris* 248 (1959), 3522—3.

[3] O. Ore, *Theory of Graphs* (Amer. Math. Soc. Coll. Pub. Vol XXXVIII, 1962).

[4] H. I. Scoins, "The number of trees with nodes of alternate parity", *Proc. Cambridge Philos. Soc.* 58 (1962), 12—16.

Department of Mathematics
University of Malaya
Kuala Lumpur, Malaysia