February, 1984

ON SCHEDULING THE CONSTRUCTION OF A TREELIKE COMMUNICATION NETWORK*

Amos Israeli[1] and Yehoshua Perl[2]

DCS-TR-141

[1]Department of Computer Science
The Technion
Haifa, Israel

[2]Department of Computer Science
Rutgers University
New Brunswick, New Jersey

and

Bar-Ilan University
Ramat-Gan, Israel

Department of Computer Science
Hill Center for the Mathematical Sciences
Busch Campus
Rutgers University
New Brunswick, New Jersey 08903

---

*This work was done while the first author was a student in the Weizmann Institute of Science Rehovot, Israel.

ABSTRACT

In a treelike communication network, transmission equipment is required for the internal vertices. We consider optimal schedules for the construction of the edges of the network, minimizing the cost of hiring the transmission equipment during the construction process.

Properties of optimal schedules are investigated. It is shown that an optimal schedule is constructed in three parts: The initial matching, the stars part and the residual subgraph. Rules concerning the construction of each one of the parts are presented. Efficient algorithms are presented for some special cases of paths of stars.

1. INTRODUCTION

Consider a communication network of a treelike structure. Communication between any two vertices in the network is transferred through the unique path connecting these two vertices. If the two vertices are not adjacent, then the other vertices on the path connecting the two vertices are used as transmission vertices. Hence every internal vertex of the network requires transmission equipment (e.g., a directory), which is more expensive than regular communication equipment. Thus, the cost of such a communication network depends on the number of internal vertices in the tree. This fact motivated Zelinka [Z] in looking for a spanning tree with the minimum number of internal vertices of a given graph. He presented an exponential algorithm for the problem. Garey and Johnson [GJ], show that the corresponding decision problem is NP-Complete.

Consider now the scheduling of the construction of the edges of a tree-like communication network T. We assume that the transmission equipment for the network is hired, with a constant fee for hiring one unit of transmission equipment per time unit. The total cost of hiring the transmission equipment during the construction of the network depends on the number of internal vertices in each stage of the construction. Our purpose is to find a schedule for the construction of the network minimizing the total cost of hiring the transmission equipment during the construction. Such a schedule is an <u>optimal schedule</u>. (Note that the cost of hiring the transmission equipment after the network is constructed is independent of the schedule of the construction.)

In [PY] it is assumed that the network is constructed in a schedule such that at each stage of the construction, a maximum number of pairs of vertices can communicate with each other. Thus the edges constructed up to each step generate a (connected) subtree of T. Under this assumption the construction scheduling problem was reduced to the known weighted mean flow scheduling problem with tree-like precedence constraints [Ho], [AH], [A].

In this work we consider the case where the connectivity requirement is released. The higher degree of freedom in this case enables schedules of cheaper total hiring cost. On the other hand, it increases the complication of the problem. Because of this complication, we assume in this work that the construction of each edge of the network takes one unit of time.

We proceed by introducing several definitions and notations. Throughput this work we consider undirected trees of n edges using the usual terminology of Graph Theory (see e.g., [Ha]). A schedule for construction the edges of a tree $T = (V,E)$, abbreviated as <u>a schedule of a tree T</u>, is a one to one function $\alpha: \{1,\dots,n\} \to E$.

Given a schedule α, a vertex u is called a <u>type-0</u>, <u>type-1</u>, <u>type-2 vertex</u> at stage k of the schedule, if it is incident with 0,1 or at least 2 of the edges $\alpha_1,\dots,\alpha_k$, respectively.

Let α be a schedule of T. The <u>k-th stage cost</u> N_k, $0 \le k \le n$, of α is the number of type 2 vertices at stage k (i.e., the number of internal vertices of the edge subgraph induced by $\{\alpha_1,\alpha_2,\dots,\alpha_k\}$). Clearly $N_0 = N_1 = 0$.

The <u>k-th stage difference</u> $1 \le k \le n$, is $d_k = N_k - N_{k-1}$.

The total hiring cost of a schedule α, abbreviated as the cost of α is $c(\alpha) = \sum_{j=1}^{n} N_j$. It is easy to show that $c(\alpha) = \sum_{j=1}^{n} (n-j+1)d_j$.

A schedule α is an optimal schedule if there is no other schedule α' such that $c(\alpha') < c(\alpha)$.

An edge e is a 0-edge, 1-edge, or 2-edge at stage k of a schedule α if its construction at stage $k+1$ addes 0,1 or 2 to N_k, respectively.

Let α_{i+1} be the first edge adjacent to a previous edge in a schedule α. Then the set of edges $\{\alpha_1,\alpha_2,\ldots,\alpha_i\}$ is a matching of T [B], called the initial matching of α (abbreviated i.m. of α).

A terminal edge is an edge incident with a terminal vertex. A chain is a tree with exactly two terminal vertices called the end vertices of the chain (all other vertices of the chain are of degree 2).

A subgraph $T' = (V',E')$ of a tree T is a tail if T' is a chain, one of the end vertices of T' is a terminal vertex in T, the degree in T of the chain's internal vertices is two, and the degree in T of the other terminal vertex is either 1 or greater than 2. Note that in case of degree one of both end vertices of the tail, the whole tree is just a chain.

A star of a vertex u, of order m, at stage k of a schedule α is a subtree containing a vertex u which is not of type 0, called the center of the star and m unconstructed edges $(u,v_1),(u,v_2),\ldots,(u,v_m)$ such that v_i, $1 \le i \le m$ is either of type 0 or of type 2 at stage k.

Note that during the construction of the tree, edges of a star are deleted from the star while they are constructed or while their other end vertex becomes a type 1 vertex. On the other hand, unconstructed edges incident with a center of a star are added to the star while their other end vertex becomes a type-2 vertex.

We continue the introduction by demonstrating one aspect of the complication of the problem. One would guess that an optimal schedule starts with an i.m., which is a maximum matching, keeping the stage-costs N_j with value zero as long as possible. This is true for many examples, but not true for the tree T given in Figure 1.

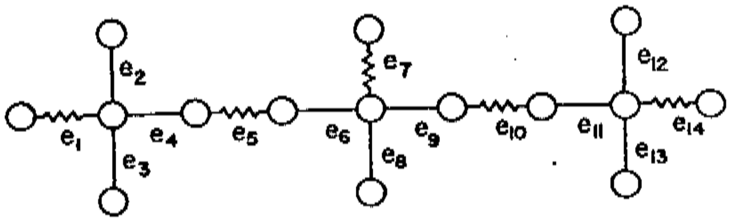


Figure 1

The set $M = \{e_1, e_5, e_7, e_{10}, e_{14}\}$ marked in Figure 1 with zigzag lines is a maximum matching of T. A cheapest schedule starting with M as the initial matching, is

$\alpha_1 = (e_1, e_5, e_7, e_{10}, e_{14}, e_2, e_3, e_{12}, e_{13}, e_{11}, e_4, e_8, e_6, e_9)$ and $c(\alpha_1) = 31$.

However, a cheapest schedule starting with the i.m. $M_1 = \{e_1, e_7, e_{14}\}$ containing the set of all terminal edges in M

is $\alpha_2 = (e_1,e_7,e_{14},e_2,e_3,e_4,e_6,e_8,e_9,e_{11},e_{12},e_{13},e_5,e_{10})$ and $c(\alpha_2) = 30$.

One may come now to the conclusion that an i.m. of an optimal schedule is either a maximum matching of T or the subset of terminal edges of a maximum matching of T. Our example demonstrates that even this conclusion is false, since an optimal schedule α_3 of T, $\alpha_3 = (e_1,e_5,e_7,e_{14},e_{11},e_{12},e_{13},e_8,e_9,e_2,e_3,e_4,e_6,e_{10})$ with cost $c(\alpha_3) = 29$ starts with the i.m. $\{e_1,e_5,e_7,e_{14}\}$.

The general structure of an optimal schedule is investigated in Sections 2 and 3. In Section 2 we study the structure of the i.m. of an optimal schedule of a tree. The construction order of stars in an optimal schedule is considered in Section 3. At the end of this section, we summarize the general results about the structure of an optimal schedule, partitioning the schedule into three different parts.

In the latter sections, we investigate some special cases. The cases of external star trees, unit distance external star trees and paths of stars are considered in Sections 4, 5 and 6, respectively, Algorithms for solving some of these special cases are presented in Section 6 and their complexity is analyzed.

In spite of all the general properties of an optimal schedule, which we found we could not solve the general case. Actually, we suspect that even in the special case of unit distance external star trees, the problem is NP-Complete. However, these properties can help the design of good heuristic methods for the problem.

2. THE INITIAL MATCHING

In this section we investigate the structure of the i.m. of an optimal schedule of a tree. Most proofs in this section are of the same nature. In order to prove that an optimal schedule has a certain property we show that for every schedule α, not having this property there exists a cheaper schedule α'. Note that α' itself is not necessarily an optimal schedule.

A matching M is maximal if no matching is obtainable by adding any edge to M. The tails subgraph TL of a tree T, is the subgraph containing all the tails of T.

Lemma 1: The i.m. of an optimal schedule of a tree, induced over the tails subgraph TL, is a maximal matching.

Proof: Like most of the following lemmas and theorems, Lemma 1 is proved by showing that for any schedule α not having this property there exists another schedule α' such that $c(\alpha') < c(\alpha)$. The proofs are technical and long and are omitted for brevity.

A matching of a tail is shifted towards the terminal vertex if the matching contains the terminal edge of the tail and no edge of the matching can be replaced by the adjacent edge closer to the terminal vertex. In case that the tree is a chain we refer to one of the end edges of the chain, arbitrarily.

A matching of a tree is terminal shifted if the matching induced over every tail of the tree is shifted towards the terminal vertex of the tail.

Lemma 2: There exists an optimal schedule α with a terminal shifted i.m.

Theorem 1: The i.m. induced by an optimal schedule over the tails subgraph TL is a maximum, terminal shifted matching.

Proof: Assume to the contrary that the i.m. induced by an optimal schedule over the subgraph TL is not a maximum matching. By Berge's Theorem [B] there exists an alternating augmenting path in TL enabling to increase the cardinality of the matching by one. This alternating augmenting path is contained either in one tail or in a chain composed of two tails having a common vertex. In both cases Lemmas 1 and 2 imply that changing the i.m. according to the alternating augmenting path, by first shifting each edge of the matching to its unmatched adjacent edge towards the terminal vertex of the tail containing this edge and then adding the last unmatched edge of the path yields a cheaper schedule, a contradiction. By Lemma 2 we may assume that the i.m. of α is terminal shifted.

Theorem 2: Let α be an optimal schedule with the i.m. $\alpha_1,\dots,\alpha_i$ such that $\alpha_1,\alpha_2,\dots,\alpha_i,\alpha_k$ for some k, $k > i+1$ is also matching. Then no vertex incident with $\alpha_k = (u,v)$ is of type 0 at stage $k-1$.

A tail is even (odd) if it consists of an even (odd) number of edges. Note that the case where the whole tree is a chain is implicitly solved by Theorem 1, and is not considered here. The end vertex of a tail which is not a terminal vertex of the tree is the connecting-vertex of the tail and the edge of the tail incident with the connecting vertex of the tail is the connecting edge of the tail. Counting the edges of a tail starting from its connecting edge, we get the odd and even edges of the tail.

Theorem 3: Let L be an odd tail of a tree T with the connecting vertex v, and the connecting edge e = (v,u). If α is an optimal schedule of T then:

a) The edge e behaves like a terminal edge, i.e., it is either in the i.m. of α or it is an edge of the star v when the construction of the i.m. is completed.
b) The odd edges of L, except e, are in the i.m. of α.
c) Each of the even edges of L is a separate 2-edge upon completion of the construction of the i.m. of α.

Proof: a) By Theorem 1, the i.m. of α induced over the tails subgraph TL of T, is a maximum terminal shifted matching. Hence, an edge of the i.m. is incident with the vertex v. If the edge e is not in the i.m. of α then it is an edge of the star of v existing upon completion of the i.m. of α. Let e' be the edge adjacent to e in L. It can be shown that e is constructed before e'.
b) Trivially implied by Theorem 1.
c) Each of the even edges of L, except perhaps the edge e', adjacent to e, is a separated 2-edge just upon completion of the i.m. of α. Thus it is constructed as a 2-edge. As shown on part a), the edge e' is constructed in α after the edge e and thus it is also constructed as a 2-edge. □

Lemma 3: Let L be an even tail of a tree T with the connecting vertex v and the connecting edge e = (v,u). There is an optimal schedule α of T such that e is constructed only after all edges incident with v which are not connecting edges of even tails are constructed.

Lemma 3 implies the following theorem:

Theorem 4: Let L be an even tail of a tree T with the connecting vertex v and the connecting edge e = (v,u). If α is an optimal schedule of T then:

a) The edge e = (v,u) is constructed either as an edge of the star of u whose order is one or as a 2-edge.

b) The even edges of L are in the i.m. of α.

c) Each of the odd edges of L, except e, is a separate 2-edge upon completion of the construction of the i.m. of α.

3. STARS

A schedule is a starry schedule if for each star all the edges of the star are constructed one by one immediately after the center of the star becomes a type-2 vertex.

Theorem 5: Let α be an optimal schedule with the i.m., $\alpha_1,\alpha_2,\ldots,\alpha_i$. Every 0-edge $\alpha_k = (u,v)$, $k > i+1$, is an edge of a star.

Proof: Since α_k is a 0-edge both u and v are either of type 0 or of type-2 at stage $k-1$. By Theorem 2, at least one of the two vertices say u, is a type-2 vertex at stage $k-1$, since if both vertices are of type 0 at this stage, then $\alpha_1,\alpha_2,\ldots,\alpha_i,\alpha_k$ is a matching. Hence $\alpha_k = (u,v)$ is an edge of the star of u at stage $k-1$.

□

Lemma 4: A center of a star becomes a type 2 vertex in an optimal schedule α, by constructing an edge of the star.

Lemma 4 implies that a schedule is starry if for each star all the edges of the star are constructed immediately after the first edge of the star is constructed.

Theorem 6: An optimal schedule is a starry schedule.

For a given schedule α, of the tree T we define the $i - j$ subschedule of α, $1 \leq i < j \leq n$, as the sequence of the edges $\alpha_i\alpha_{i+1},\ldots,\alpha_j$. Let β be the $i - j$ subschedule of α. The subgraph of β is the subgraph induced by $\alpha_i,\alpha_{i+1},\ldots,\alpha_j$. The cost of β is is $c(\beta) = \sum_{k=i}^{j} N_k$. A j-prefix of a schedule α is the $1 - (j-1)$

subschedule of α. The k-suffix of a schedule α is the (k+1) - n subschedule of α.

Lemma 5: Let β and β' be the i - k subschedules of two schedules α and α', respectively, whose i-refix and k-suffix are the same. If $c(\beta) > c(\beta')$ then $c(\alpha) > c(\alpha')$.

For a given schedule or subschedule, we define its stage differences sets $D_i = \{j | d_j = i\}$, $i = 0,1,2$.

Lemma 6: Let β and β' be the i - k subschedules of two schedules α and α', respectively, whose i-prefix and k-suffix are the same. Let the stage differences sets of α and α' be D_i and D_i', $i = 0,1,2$, respectively. If the conditions a, b and c hold, then $c(\beta) < c(\beta')$.

a) $D_o = D_2 = \emptyset$

b) $|D_o'| = |D_2'| > 0$

c) There is an one to one correspondence between D_o' and D_2' such that every 0-edge of β' is preceded by the corresponding 2-edge of β'.

Proof: Since both schedules have the same i-prefix $N_{i-1} = N_{i-1}'$. For j, $i \leq j \leq k$, $N_j = N_{i-1} + (j-i+1)$ since $d_i = d_{i+1} = \dots = d_j = 1$. Condition (c) implies that for every j, $i \leq j \leq k$, the number of 2-stage differences between stages i and j is greater than or equal to, the number of 0-stage differences between stages i and j. Thus, for every j, $i \leq j \leq k$, $N_j' \geq N_{i-1} + (j-i+1) = N_j$. Let ℓ, $i \leq \ell < k$ be the first index such that $d_\ell = 2$. Then $d_j' = 1$ for j $i < j < \ell$ and thus $N_j = N_j'$ for j, $i \leq j < \ell$ and $N_\ell' = N_\ell + 1$, Hence $c(\alpha) < c(\alpha')$. □

Theorem 7: Let G be a subtree which is the subgraph of the i - k subschedule β of an optimal schedule α. If all the edges of G, at stage i-1, are 2-edges except a single 1-edge e, then the i - k subschedule β, of α, starts with e and proceeds by constructing G in a connected manner.

Proof: Assume to the contrary the existence of an i - k subschedule β' of α, not satisfying the requirement of the theorem. Let D_i and D_i', i = 1,2,3, be the stage difference sets of β and β', respectively. Clearly, $D_o = D_2 = \emptyset$. Thus, $|D_o'| = q > 0$.

At stage i-1 all the vertices of G are of type 1 except one vertex incident with e, which is either of type 0 or of type-2. The first 0-edge of β' is incident with two type-2 vertices which belong to different constructed components of the tree G, since otherwise G contains a cycle. At most, one of these constructed components contains the edge e. Thus, the first constructed edge of the other component is incident with two vertices of type 1 and is a 2-edge, constructed before the first 0-edge of β'.

In a similar way, it can be shown that for each of the q, 0-edges of β' corresponds a distinct 2-edge constructed before the corresponding 0-edge. In β, $|D_1| = k-i+1$, and thus $|D_2'| = |D_o'| = q$ since $N_{i-1}' = N_{i-1}$ and $N_k' = N_k$. Thus by Lemma 6, $c(\beta) < c(\beta')$. Therefore, Lemma 5 implies that β' is not an i - k subschedule of α since α is an optimal schedule.

□

Theorem 8: Let G be a subtree which is the subgraph of the i-k subschedule β of an optimal schedule α. If all the vertices of G are of type 1 at stage i-1, then β starts with an arbitrary edge of G and proceeds constructing G in a connected manner.

Proof: Since all the edges of the subtree T are 2-edges, the construction starts with a 2-edge. Then we have either one or two connected subtrees, satisfying the conditions of Theorem 7. By this theorem we can continue the construction as described.

□

Lemma 7: Let α_k be a 2-edge in an optimal schedule α. The first edge, in α, after α_k which is not a 2-edge at stage k-1, is a 1-edge at stage k-1.

Proof: Let $\alpha_m = (u,v)$ be the first edge in α, after α_k, that is not a 2-edge at stage k-1. Assume to the contrary that α_m is a 0-edge at stage k-1. The vertices u and v are of type-0 or type-2 at stage k-1.

Case 1: One of the two vertices, say u, is of type 2 and the other vertex v of type 0 at stage k-1.

The edge (u,v) belongs to the star of u at this stage. The vertex u is of type-2, hence the construction of the star of u which starts at an earlier stage is interrupted. By by Theorem 6, the schedule α is starry, a contradiction.

Case 2: Both vertices are of type 2 at stage k-1.

The contradiction is obtained similarly.

Case 3: Both vertices are of type 0 at stage k-1.

Theorem 2 implies that both u and v are not type-0 vertices at stage m-1. Without loss of generality, we assume that u becomes a type-1 vertex before v does, by construction of an edge α_j, $k < j < m$, incident with u. But α_j is not a 2-edge at stage k-1, contradiction to our assumption about α_m being the first such edge.

□

A schedule is <u>greedy</u> if it is starry and no 2-edge is constructed while a 0-edge or a 1-edge can be constructed.

<u>Theorem 9</u>: An optimal schedule is greedy.

<u>Proof</u>: Assume, to the contrary that α_k is a 2-edge while the construction of an edge which is not a 2-edge at stage k-1 is delayed.

Let α_m, $m > k$, be the first edge in α, that is, not a 2-edge at stage k-1. By Lemma 7, α_m is a 1-edge at stage k-1. Let β be the k-m subschedule of α with the subgraph G. We shall present a cheaper k-m subschedule β' of G, contradicting the optimality of α by Lemma 5. Consider the following cases:

<u>Case 1</u>: The edge α_m is a separate component of G.

All the vertices of $G - \alpha_m$ are type-1 vertices at stage k-1. By Theorem 8, an optimal $k - (m-1)$ subschedule of $G - \alpha_m$ starts with a 2-edge and continues with a sequence of 1-edges and 2-edges where the number of 2-edges is the number of components of G. One can see that the $k - m$ subschedule of G, $\beta' = (\alpha_m, \alpha_k, \alpha_{k-1}, \ldots, \alpha_{m-1})$ satisfies $c(\beta') < c(\beta)$.

<u>Case 2</u>: The edge α_m is not a separate component of G.

Consider the connected component G' of G, containing α_m. At stage $i-1$, the edges of G' are all 2-edges except the 1-edge α_m. By Theorem 7, an optimal construction of G' starts with construction of α_m and continues in a connected manner. It can be shown that a subschedule β' starting with this construction of G', and proceeding like β satisfies $c(\beta') < c(\beta)$. □

Let α_k be the first 2-edge of an optimal schedule α with the i.m., $\alpha_1, \alpha_2, \ldots, \alpha_i$. We call the $(i+1)$ - $(k-1)$ subschedule <u>the star construction phase of α</u>. The subgraph of the $(k-1)$-suffix of α is the <u>residual subgraph of α</u>.

<u>Lemma 10</u>: Let G be the residual subgraph of an optimal schedule whose first 2-edge is α_k. The vertices of G are all of type 1 at stage $k-1$.

<u>Proof</u>: By Theorem 9, the optimal schedule α is greedy. Assume to the contrary that G has a vertex which is not of type 1 at stage $k-1$, then G contains either a 1-edge or a 0-edge incident with this vertex. But the construction of such an edge precedes the construction of the 2-edges α_k since the schedule α is greedy, a contradiction. □

<u>Theorem 10</u>: Let $G_i = (V_i, E_i)$, $i = 1,2,\ldots,r$ be the connected components of the residual subgraph G of an optimal schedule α. Then the connected components G_i are constructed in α, one by one, in a decreasing order of $|V_i|$.

<u>Proof</u>: By Theorem 9, α is a greedy schedule. Thus by Lemma 9, all the vertices of the residual subgraph G are of type 1. The first

constructed edge of G is a 2-edge. Once this edge is constructed, it turns at least one edge in its component to a 1-edge (provided that the corresponding E_i satisfies $|E_i| > 1$). Since α is greedy, this 1-edge is constructed before any 2-edge. Thus, all of the edges of this component are constructed (in a connected manner as shown in Theorem 8), before any edge of another component is constructed, since the first constructed edge of a component is a 2-edge. Hence, the residual subgraph is constructed component by component. It is easy to see that a non-increasing order of $|V_i|$ yields the cheapest cost. □

For conclusion we summarize the main results of our investigation.

The construction of an optimal schedule is composed ot three phases:

1) The initial matching phase; the stage differences are all equal to zero.

2) The stars construction phase; the stage differences are either zero or one.

3) The residual subgraph construction phase; the stage differences are either one or two.

The i.m. over the tails subgraph is a maximum terminal shifted matching. However, the structure of the i.m. over the other parts of the tree is not entirely clear.

An optimal schedule is starry, i.e., the stars are constructed one at a time, such that all the edges of a star are constructed one by one with no interruption. The first edge of a star is a 1-edge and all the other edges are 0-edges. The order of constructing the different stars in an optimal schedule is still to be determined.

The components of the residual subgraph are constructed one by one in a non-increasing order of their sizes. The construction of each component starts with an arbitrary 2-edge and proceeds in a connected manner with 1-edges.

4. EXTERNAL STAR TREES

A tree T is an external star tree if every vertex of T of degree greater than two is incident with a terminal edge.

Two stars, S_1 and S_2 of a tree T whose centers v_1 and v_2 are connected by a chain (i.e., there is no third star between them), are neighbors. If the length of the chain is one, i.e., $(v_1,v_2) \in E$, then S_1 and S_2 are adjacent.

The star order sequence of a greedy schedule α is a sequence of integers s_j, $1 \leq j \leq r$, where s_j is the order of the j-th star, constructed in α, just before starting its construction. The star construction cost of a greedy schedule α, is the sum $SC(\alpha) = \sum_{j=1}^{m} N_j$ where α_{m+1} is the first 2-edge constructed in α. Note that if α and α' are two greedy schedules with the same residual subgraph, then $SC(\alpha) < SC(\alpha')$ implies that α' is not optimal. A sequence of integers $a_1,a_2,\ldots,a_k$ is almost decreasing if for every i and j, $1 \leq i < j \leq k$, $a_i+1 \geq a_j$.

Lemma 11: Let $(s_1,s_2,\ldots,s_k)$ be the star order sequence of a greedy schedule α. The star construction cost of α satisfies $SC(\alpha) = \sum_{j=1}^{r} j \cdot s_j$.

Proof: This formula is implied from the fact that the schedule α is starry and the following observation: the only stage difference of value one during the construction of a star is at the stage at which the first edge of the star is constructed.

□

Lemma 12: Let $(s_1,s_2,\ldots,s_r)$ be the the star order sequence of an optimal schedule α of an external star tree T where s_j, $1 \leq j \leq r$

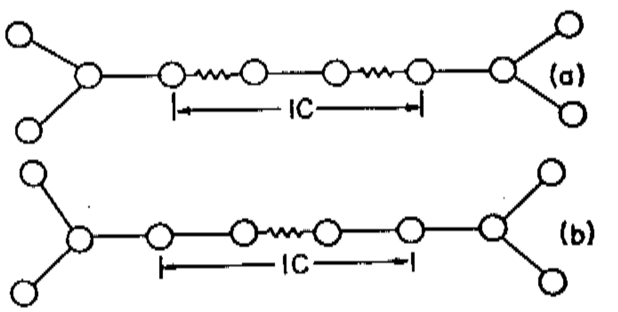
is the order of the star of u_j. Then

a) For every j, $1 < j \leq r$, $s_j \leq s_{j-1} + 1$.

b) If $s_j = s_{j-1} + 1$, then the stars of u_{j-1} and u_j are adjacent and have the same order just before constructing the star of u_{j-1}.

<u>Theorem 11</u>: The star order sequence of an optimal schedule of an external star tree is an almost decreasing sequence.

Let C be a chain of more than two edges in a tree T, connecting the centers u and v of two stars, whose other vertices are of degree two in T. The terminal edges of C are (u,u_1) and (v,v_1). The chain $IC = c - \{(u,u_1),(v,v_i)\}$ is the <u>internal chain</u> of u and v (see Figure 2).



(a) odd matching (b) even matching

<u>Figure 2</u>

The edges (u,u_1) and (v,v_1) are the connecting edges of IC. An internal chain is even (odd) if it consists of an even (odd) number of edges. Let IC be an odd internal chain. Counting its edges from any of the terminal edges, we get the odd and even edges of IC. A maximal matching of IC is even (odd) if it consists of all the even (odd edges of IC) (see Figure 2).

Let IC be the even internal chain of u and v. A maximum matching of IC is u-justified if it consists of the terminal edge of IC closer to u and every second edge from it (see Figure 3).

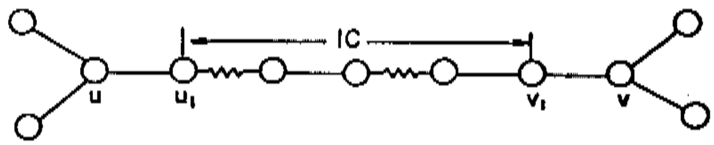


Figure 3

Lemma 13: Let IC be an internal chain of a tree T containing more than one edge. The i.m. of an optimal schedule α of T induced over IC is a maximal matching.

Lemma 14: Let IC be an internal chain of a tree T and let α be a schedule of T. Let α_k be an edge of IC which is in the i.m. of α. If α_m is a non-terminal edge of IC adjacent to α_k and not adjacent of any other edge in the i.m. of α, then there exists a schedule α' of T satisfying $c(\alpha') \leq c(\alpha)$ such that α_m replaces α_k in the i.m. of α'.

Theorem 12: Let α be an optimal schedule of an external star tree T. Let IC be an internal chain of length greater than one in T.

a) If IC is an odd internal chain then the i.m. induced over IC is either even or odd.

b) If IC is an even interal chain of u and v then the i.m. over IC is v-justified in case the star of u is constructed in α before the star of v.

A condensed external star tree is obtained from an external star tree by eliminating the edges of the even internal chains and connecting the two "connecting edges" of each even internal chain to one another, and replacing each of the odd internal chains by one edge.

Corollary 1: Let T_1 be the condensed external star tree of an external star tree T. Let α be an optimal schedule of T. Then there exists an optimal schedule α_1 of T_1 conserving the star order construction in α such that $SC(\alpha_1) = SC(\alpha)$.

5. UNIT DISTANCE EXTERNAL STAR TREES

A unit distance external star tree, abbreviated as a unit distance tree is an external star tree whose vertices which do not belong to tails are all of degree not less than three. Note that the distance between the centers of any two neighbor stars of a unit distance tree is one (i.e., they are adjacent). Theorem 1 implies the following corollary:

Corollary 2: Let α be an optimal schedule of a unit distance tree. The i.m. of α is a maximum terminal shifted matching and all its edges are edges of tails.

Let α be a greedy schedule (of a tree T) whose star order sequence is $(s_1,s_2,\ldots,s_r)$. The maximum order sequence of α $(m_1,m_2,\ldots,m_r)$ is defined as follows: m_1 is the order of the maximal unconstructed star in T upon completion of the i.m. of α, and m_j, $1 < j \leq r$, is the maximal order of an unconstructed star in T just after construction of the (j-1)-th star of α. A greedy schedule of the tree T is a maximum order schedule if $m_j = s_j$, $1 \leq j \leq r$.

Theorem 13: Let α be an optimal schedule of a unit distrance tree T. Then α is a maximum order schedule.

Let T be a unit distance tree. A set of stars ST = $\{S_1,S_2,\ldots,S_r\}$ is connected if the subgraph induced by the centers of the stars is a connected subgraph. A set of stars ST is adjacent to a star S if there is a star $S' \in ST$, such that S and S' are adjacent.

A connected set of stars ST is a p-star set at stage j of a schedule α, abbreviated as a p-star set if every star of ST is of order p at stage j. A p-star set ST is maximal if no star adjacent to ST is of order p.

Lemma 15: Let α be an optimal schedule of a unit distance tree T and let ST be a maximal p-star set just before construction of the first star of ST in α. Then the stars of ST are constructed in α one after the other in a connected manner and their order in α is p+1 except for the first star of ST whose order is p.

Let ST1 and ST2 be two maximal p-star sets at stage j of a schedule α. The sets ST1 and ST2 are close if they are both adjacent to a star S, $S \notin ST1$, $S \notin ST2$, of order p-1 at stage j. The sets ST1 and ST2 are united at stage j+1 if they are close at stage j and the star S becomes of order p at stage j+1 by construction of a star S', $S' \notin ST1$, $S' \notin ST2$, adjacent to S.

The power of a set of stars is the number of the stars in it. A greedy schedule α of a unit distance tree T is a maximum power schedule if no p-star set of power r is constructed while there is an unconstructed p-star set whose power is greater than r.

A path of stars is a set of stars $ST = \{S_1, S_2, \ldots, S_r\}$ such that each star S_j, $1 < j < r$, is a neighbor of exactly two stars, S_{j-1} and S_{j+1}. The length of ST is its power

Let α' be a schedule of a unit distance tree T. A p—(p-1)—p path of stars at stage j, abbreviated as a p—(p-1)—p path is a subtree of T which is a path of stars $ST = \{S_1, S_2, \ldots, S_r\}$ such that the order of S_1 and S_r at stage j of α is p and the order of S_j, $1 < j < r$ is p-1.

Note that since ST is a subtree of T, the stars S_j and S_{j+1}, $1 \leq j < r$, are adjacent. Also note that in a maximum order schedule, a p—(p-1)—p path is optimally constructed by starting from one end of the path and continuing in a connected manner. In this case, all the stars of the path are order p while constructed except the last one, which is of order p+1.

A greedy schedule α is a shortest path schedule if no p—(p-1)—p path, p > 0, of length r is constructed while there is a shorter unconstructed p—(p-1)—p path.

A p—(p-1)—p path ST is cut if some of its stars, but not all of them, are constructed as a part of another p—(p-1)—p path. A maximum order schedule of a unit distance tree T is regular if it is a maximum power schedule and a shortest path schedule.

Theorem 14: Let α be an optimal schedule of a tree of unit distance T. If no p-star sets are united and no p — (p-1)—p path is cut, then α is a regular schedule.

Proof: Note that the maximum power rule implies that a p—(p-1)—p path is constructed only if there are no unconstructed p-star sets of power greater than one. It can be shown that if two p-star sets are united in a certain maximum order schedule of T, then some sets will be united in any maximum order schedule of T. Furthermore, it can be shown that if a p—(p-1)—p path is cut in a certain maximum order schedule of T, then some p—(p-1)—p path is cut in any maximum order schedule of T. Thus, occurrances of a unification of p-star sets and cut paths are properties of the tree T and not of the maximum order schedule. By Theorem 13, α is a maximum order schedule of T. Hence, by our assumptions no occurrance of unification of p-star sets or cut $p-(p-1)-p$ paths is possible in any maximum order schedule of T. By Lemma 15, p-star sets are constructed as a whole. By simple transformations it can be shown that if α' is a maximum order schedule of T which is not a regular schedule, then there is a maximum order schedule α'' of T such that $c(\alpha'') < c(\alpha')$. Thus the theorem is proved by contradiction.

□

The k p-star sets $ST_1, ST_2, \ldots, ST_k$ are close at stage j of a schedule α if for each two of them ST_i and ST_j, $1 \le i \le j \le k$, there is a sequence of sets $ST_i = ST_{i_1}, ST_{i_2}, \ldots, ST_{i_\ell} = ST_j$ such that for m, $1 < m \le \ell$, the sets ST_{i_m} and $ST_{i_{m+1}}$ are close.

A regular schedule of a unit distance tree T is a maximum close sets schedule if whenever a p-star set of power m is to be constructed, a p-star set in the largest set of close sets is constructed. Two $p-(p-1)-p$ paths $ST = \{S_1, S_2, \ldots, S_r\}$ and $ST' = \{S_1', S_2', \ldots, S_s'\}$ are linked if $S_r = S_1'$. The k $p-(p-1)-p$ paths $SR_1, ST_2, \ldots, ST_k$ are linked if for every two paths ST_i and ST_j, $1 \le i < j \le k$ there is a sequence of paths $ST_i = ST_{i_1}, ST_{i_2}, \ldots, ST_{i_\ell} = ST_j$ such that for m, $1 \le m < \ell$ the paths ST_{i_m} and $ST_{i_{m+1}}$ are linked.

A regular schedule of a unit distance tree T is a maximum linked paths schedule if whenever a $p-(p-1)-p$ path of length m is to be constructed a path among the largest set of linked $p-(p-1)-p$ paths of length m is constructed.

Theorem 15: Let α be a regular schedule of a unit distance tree T. Assume that no p-star sets are united and no $p-(p-1)-p$ paths are cut in α. If α is a maximum close set schedule, and a maximum linked paths schedule, then α is optimal.

The following example demonstrates that in case of unification of p-star sets, a maximum power schedule is not optimal. In Figure 4(a), stars are represented by drawing their centers as squares. The numbers in the squares are the order of the stars after construction of the i.m. (see Figure 4(b) for the original tree after the construction of the i.m.)

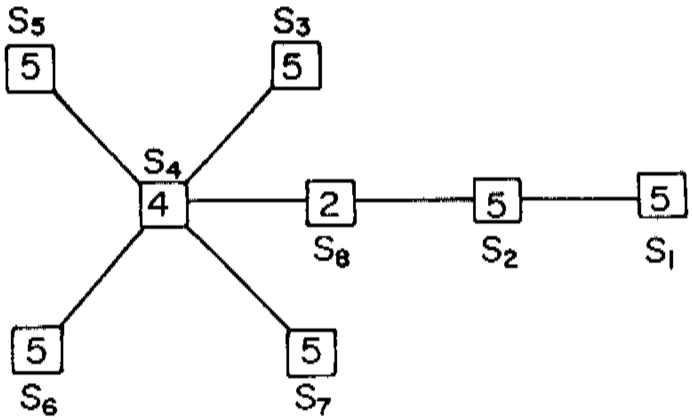


Figure 4(a)

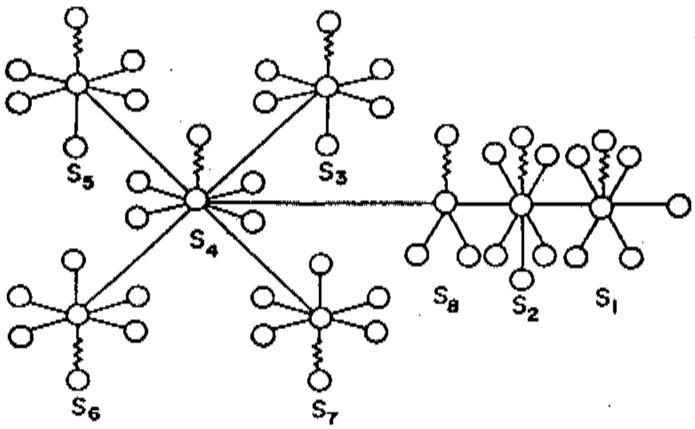


Figure 4(b)

A maximum power schedule implies the following order of stars: $S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8$. The cost of this schedule is 192. However, the cost of the scheduling: $S_3, S_4, S_5, S_6, S_1, S_2, S_8$ is only 191.

In a similar way it can be demonstrated that cutting a $p-(p-1)-p$ path, while using the shortest path rule may yield a non-optimal schedule.

6. PATHS OF STARS

A unit distance path of stars is a path of stars where each of the stars is adjacent to its two neighboring stars. An even path of stars is a path of stars where each star is connected to its neighbors by an even chain. An odd path of stars is a path of stars where each star is connected to its neighbors by an odd chain of length greater than one.

Theorem 16: Let T be a unit distance path of stars. A regular schedule α of T which is a maximum close sets schedule and a maximum linked paths schedule is an optimal schedule of T.

Proof: Since T is a path of stars, each star of T is adjacent to at most two stars and thus no p-star sets may be united and no $p-(p-1)-p$ path may be cut. Thus, the schedule α satisfies the conditions of Theorem 15. Hence α is an optimal schedule of T.

□

We present now an algorithm for optimal scheduling of the construction of a unit distance path of stars.

Algorithm 1:

1. Construct the maximum terminal shifted i.m. over all tails of T;
2. While T contains unconstructed stars;

    do

    Let m be the largest order of unconstructed stars;

    Let ℓ be the largest power of an m-order set;

    If $\ell > 1$;

    then

    Construct the star sets in the largest set of close sets of power ℓ of stars of order m;

    else

```
            do
              while T contains unconstructed  m-(m-1)-m  paths;
                  do
                    Let k be the length of the shortest
                    m-(m-1)-m  path;
                    Construct a set of maximum power of linked
                    m-(m-1)-m  paths of length  k.
                  end
              Construct the unconstructed (isolated) m-order star;
        end;
    end;
```

3. Construct the residual subgraph containing the isolated unconstructed edges of the tails.

By Theorem 16 the algorithm yields an optimal schedule. A careful analysis of the complexity of an efficient implementation of Algorithm 1 (using priority queues implemented as heaps) yields an O(max(n,s log s)) complexity where s, $s < n$, is the number of stars in the tree.

We demonstrate now Algorithm 1 for the tree in Figure 5. Figure 5 uses the conventions introduced in Figure 4(a). The tree is drawn after the i.m. is already constructed. We bring the order of the stars

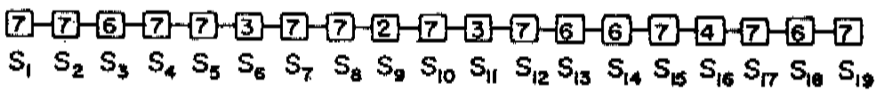


Figure 5

as shown in Algorithm 1. The largest order of a star is 7. There are three, 7-star sets of power two. The sets $\{S_1,S_2\}$ and $\{S_4,S_5\}$ are close and thus one of them, say S_1,S_2, is constructed first. The order of S_1, while constructed is 7; the order of S_2, while constructed is 8. Now the set $\{S_3,S_4,S_5\}$ is constructed next where the order of the stars S_3, S_4 and S_5 are 7, 8 and 8 respectively. The only remaining 7-star set of power two is $\{S_7,S_8\}$ and now it is its turn. Then the only remaining 7-star sets are of power one so we look for $7-(6)-7$ paths. There are no such linked paths in this example. The shortest $7-(6)-7$ path is $\{S_{17},S_{18},S_{19}\}$ and it is constructed next starting from one end, say S_{17}. Both orders of S_{17} and S_{18} while constructed are 7 while S_{19} is of order 8 when constructed. Then the longer $7-(6)-7$ path $\{S_{12},S_{13},S_{14},S_{15}\}$ is constructed. Then S_{10} the only isolated star of order 7. The remaining stars are S_{16}, S_{11}, S_6 and S_9. Their orders at this stage are 6, 5, 5 and 4, respectively, and they are constructed in this order. The residual subgraph in this example is empty.

Theorem 17: Let T be an even path of stars. If α is an optimal schedule of T, then α is a maximum order schedule.

Let T be an even path of stars. A maximum order schedule α of T is an end justified schedule if whenever it constructs a set ST of k, $k > 2$, neighboring stars of order m, it starts from an end star of ST.

Theorem 18: Let T be an even path of stars. A maximum order schedule α of T, which is an end justified schedule is an optimal schedule.

We present now an algorithm for finding an optimal schedule of an even path of stars.

Algorithm 2:

1. Construct the maximum terminal shifted i.m. over all the tails of T;
2. While T contains unconstructed stars;

   do

   Let m be the largest order of an unconstructed star;

   Construct an end star S of order m;

   If any star S_1, which is a neighbor of S is not constructed

   then

   Add to the i.m., constructed at stage 1, the i.m. over the internal chain between S and S_1 justified towards the center of S_1;

   end;
3. Construct the residual subgraph.

By Corollary 1 and Theorem 18, Algorithm 2 yields an optimal schedule. Note the order of the stars of T only decreases during the process and thus the star order sequence obtained is a decreasing sequence. Note also that if IC is the even internal chain of u and v, and if the star of u is constructed before the star of v, then the i.m. of IC is v-justified and the following happens:

a) The connecting edge of IC incident with the vertex u is constructed as an edge of the star of u.

b) The connecting edge of IC incident with the vertex v is adjacent to an edge of the i.m. and in the case IC is null, it is also adjacent to the connecting edge incident with u. In both cases this connecting edge becomes, after construction of the star of v, a star of order one and constructed at the end of the star construction phase of α.

It is easy to see that the complexity of Algorithm 2 is

O(max(n,s log s)). We demonstrate Algorithm 2 on the tree in Figure 6.

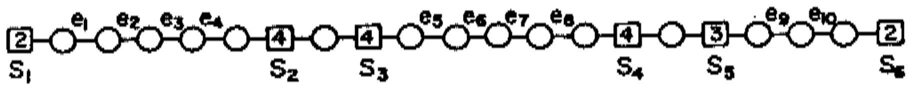


Figure 6

The scheduling of the stars obtained by Algorithm 2 after construction of the i.m. over the tails is: S_2, S_4, S_5, S_6, S_1. The choice of S_2 before S_4, S_5 before S_6 and S_6 before S_1 is arbitrary. After construction of S_2, the edges e_1 and e_3 are added to the i.m. After construction of S_4, the edges e_4 and e_7 are added to the i.m. The edge e_{10} is added to the i.m. after construction of S_5. The residual subgraph contains the edges e_2, e_4, e_6, e_8, e_9.

It may seem that an optimal schedule for a "mixed" path of stars with even lengths and length one is a maximum order schedule. The following example demonstrates it is not always true.

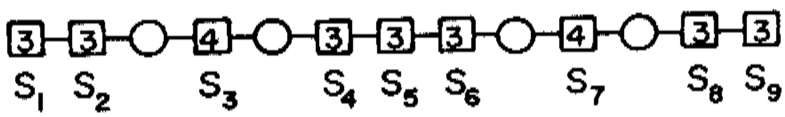


Figure 7

The cheaspest maximum order schedule of the tree in Figure 7 is $S_3,S_7,S_1,S_2,S_5,S_4,S_6,S_9,S_8$ and its cost is 138. However, the cost of the schedule $S_5,S_4,S_6,S_1,S_2,S_8,S_9,S_3,S_7$ which is not a maximum order schedule is only 135...

Theorem 19: An optimal schedule of an odd path of stars is a maximum order schedule.

Proof: Since the order of no star is changed by the construction of its neighbors the theorem is proved immediately by simple transformations.

□

Note that once the i.m. of a maximum order schedule of an odd path of stars is constructed, the schedule is fully determined. In Theorem 12(a), we proved that the i.m. induced by an optimal schedule over any odd internal chain of an external star tree T is either an even matching or an odd matching. Let α be a maximum order schedule of T. Let the i.m. induced over an internal chain IC, of T, be an even matching. The schedule α_{IC} is obtained from α by replacing the even i.m. of IC by an odd matching of IC. The profit in this interchange is $PR(\alpha, IC) = c(\alpha) - c(\alpha_{IC})$.

We conjecture that the following algorithm determines the i.m. of an optimal schedule of an odd path of stars.

Algorithm 3:

1. Construct the i.m. over all the tails of T;
2. Construct an even i.m. over all the internal chains of T;
   Let α be the maximum order schedule determined by the current i.m.
   While the internal chain IC of T maximizing PR(α,IC) satisfies
   PR(α,IC) > 0;
   Replace α by α_{IC}.

It can be shown that the complexity of Algorithm 3 is $O(\max(n,s^2))$. Note that if the above conjecture is true, then a maximum order schedule using the i.m. found by Algorithm 3 is an optimal schedule of T. Hence an algorithm for finding an optimal schedule for an odd path of stars is obtainable.

REFERENCES

[A] Adolphson, D., "Single Machine Job Sequencing with Procedure Constraints", SIAM J. Comput. 6 (1977), 40-54.

[AH] Adolphson, D. and Hu, T.C., "Optimal Linear Ordering" SIAM J. Applied Math., 25 (1973), 403-423.

[B] Berge, C., "The Theory of Graphs and its Applications" John Wiley and Sons, Inc., New York, 1962.

[GJ] Garey, M.R. and Johnson, D.S., "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman and Co., San Francisco, 1979.

[HA] Harary, F., "Graph Theory", Addison Wesley Publishing Co., Reading, Massachusetts, 1969.

[HO] Horn, W.A., "Single Machine Job Scheduling with Treelike Precedence Ordering and Linear Delay Penalties", SIAM J. Applied Math., 23 (1972), 189-202.

[PY] Perl, Y. and Yesha, Y., "Mean Flow Scheduling and Optimal Construction of a Treelike Communication Network", Networks, 11 (1981) 87-92.

[Z] Zelinka, B., "Finding a Spanning Tree of a Graph with Maximal Number of Terminal Vertices", Kybernetika, 9 (1973), 357-360.