Theorem 5: $\boldsymbol{a} = S_n f$ is a solution with polarity function

$$X_1 = (\overset{*}{x}_n, \cdots, \overset{*}{x}_{i_1}, \cdots, \overset{*}{x}_{i_2}, \cdots, \overset{*}{x}_{i_k}, \cdots, \overset{*}{x}_1)$$

if and only if

$$\boldsymbol{a}' = S_n \cdot f^{\bar{x}_{i_1}, \bar{x}_{i_k}, \cdots, \bar{x}_i}$$

is a solution with polarity function

$$X_2 = (x_n, \cdots, \bar{\overset{*}{x}}_{i_1}, \cdots, \bar{\overset{*}{x}}_{i_2}, \cdots, \bar{\overset{*}{x}}_{i_k}, \cdots, \overset{*}{x}_1).$$

We observe that Theorems 2 and 4 are special cases of Theorem 5. However, Theorems 2 and 4 result into a simple algorithm for generating all the solutions.

V. Algorithm

The steps of the algorithm are as follows.

Step 1: Write $f = (f_0, f_1, \cdots, f_{2^n - 1})^T$.

Step 2: Obtain the vector $\boldsymbol{a}$ by multiplying S^n and f.

Step 3: Obtain $g = 1/f$ and the new $\boldsymbol{a} = S^n \cdot g$. The polarity now is complement of Step 2.

Step 4: If all the solutions have been generated then stop.

Step 5: Obtain a new polarity function and set $f \leftarrow f^{\bar{x}_i}$. Go to Step 2. The correctness of each solution is guaranteed by Theorem 4. The following example explains the steps of the algorithm.

Example: $f(x_3, x_2, x_1) = \sum (1, 3, 4, 7)$. For generation of different polarities we will use Gray code, the polarity functions are used in the order (x_3, x_2, x_1), $(x_3, x_2, \bar{x}_1)$, $(x_3, \bar{x}_2, \bar{x}_1)$, $(x_3, \bar{x}_2, x_1)$. The complete solutions is given below.

S^3 as in (1). f vectors are as follows

f	$\frac{1}{f}$	$g = f^{\bar{x}_1}$	$\frac{1}{g}$	$h = g^{\bar{x}_2}$	$\frac{1}{h}$	$k = h^{\bar{x}_1}$	$\frac{1}{k}$
0	1	1	0	1	1	0	0
1	0	0	1	0	0	1	1
0	0	1	1	1	0	0	1
1	1	0	0	0	1	1	0
1	1	0	0	1	0	0	1
0	0	1	1	0	1	1	0
0	1	1	0	0	0	1	1
1	0	0	1	1	1	0	0

$\boldsymbol{a}$ vector with respective polarities are

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It is easy to see that the minimal solution is with polarity $(x_3, \bar{x}_2, x_1)$ where $a_1 = a_6 = 1$ and therefore $f = x_1 \oplus \bar{x}_2 x_3$.

Obviously the solution can be obtained sequentially and it is not necessary to retain all the f and $\boldsymbol{a}$ vectors. Certain simple observations can be included in the program implementing the algorithm, e.g., $a_{2^n - 1}$ needs to be worked out only once, based on experience a threshold number can be chosen and as soon as the total number of a_i's in any $\boldsymbol{a}$ vector exceed this threshold we can go on to next polarity function, etc.

VI. Conclusion

We have shown that Swamy's [1] solution for minimal RMC expansion is in error. A different solution has been presented to obtain the minimal solution sequentially.

Acknowledgment

The authors wish to thank Prof. Breuer and the reviewers for their suggestions which helped considerably to improve the presentation of the material.

References

[1] S. Swamy, "On generalized Reed–Muller expansions," *IEEE Trans. Comput.*, vol. C-21, pp. 1008–1009, Sept. 1972.

[2] A. Mukhopadhyay and G. Schmitz, "Minimization of EXCLUSIVE-OR and logical equivalence switching circuits," *IEEE Trans. Comput.*, vol. C-19, pp. 132–140, Feb. 1970.

[3] S. B. Marinkovic and Z. Tosic, "Algorithm for minimal polarizated polynomial form determination," *IEEE Trans. Comput.*, vol. C-23, pp. 1313–1315, Dec. 1974.

[4] F. P. Preparata, "State logic relations for autonomous sequential networks," *IEEE Trans. Comput.*, vol. EC-13, pp. 542–548, Oct. 1964.

The Design of Small-Diameter Networks by Local Search

SAM TOUEG AND KENNETH STEIGLITZ

***Abstract*—A local search algorithm for the design of small-diameter networks is presented for both directed and undirected regular graphs. In all cases the resulting graphs are at least as good as any previously known, in the sense that they have at least as small a diameter and average shortest distance for a given number of nodes and degrees.**

***Index Terms*—Communication networks, delay, diameter, graphs, networks.**

I. Introduction

Graphs with small diameters are especially well suited for designing communication networks where the elements at the vertices are very reliable and the traffic density is low, so that messages can be routed by the shortest path. Such a situation arises quite naturally in the design of networks of microprocessors; the shortest distance between two vertices then represents the delay encountered in shortest path communication between these two vertices, and the diameter is the maximum delay of this kind possible. This correspondence treats the problem of constructing graphs with small diameters.

The approach is extendable to more general problems that might arise in practical situations since additional constraints can be easily incorporated in the proposed algorithm. For example, if network reliability is an issue, a k-connectivity test can be included in the local search.

We shall consider both undirected and directed graphs $G = (V, E)$. Each edge $e \in E$ will have weight 1, and the weight of the

Manuscript received December 22, 1978. This work was supported in part by the National Science Foundation under Grant GK-42048, and in part by the United States Army Research Office, Durham, NC, under Grant DAAG29-75-G-0192.

The authors are with the Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ 08540.

shortest path from vertex i to vertex j will be denoted by d_{ij}. The *diameter* of G will be defined by

$$k = \max_{i,j \in V} d_{ij}.$$

The *degree* of a node in an undirected graph is the number of edges incident to it. In a directed graph, the *in-degree* and *out-degree* of a node are the number of edges directed in and out, respectively; if they are equal, we use the term *degree* in the directed case as well. Thus, a degree-2 directed graph has in-degree = out-degree = 2. A *regular* graph is a graph where all the nodes have the same degree.

We begin by reviewing other work in this area.

A. Moore Graphs

It is easy to see that a regular undirected graph of degree d and diameter k has at most

$$N_{um}(d, k) = 1 + d + d(d-1) + \cdots + d(d-1)^{k-1} = \frac{d(d-1)^k - 2}{d-2} \qquad (d > 2) \qquad (1)$$

nodes. The graphs achieving this upper bound are called (d, k) *undirected Moore graphs* (UMG's). UMG's have been investigated in [8], [3], and [5]. They are very rare and for nontrivial values of d and k ($d > 2$ and $k > 1$) there are only two, or possibly three, graphs achieving the upper bound (1).

Let $N_u(d, k)$ be the maximal number of nodes in a (d, k) regular undirected graph; the (d, k) graphs achieving this number are called *maximal*. At present, very few values of $N_u(d, k)$ are known (see [6]).

We can formulate similar upper bounds for directed graphs. Clearly, a regular directed graph of degree d and diameter k has at most

$$N_{dm}(d, k) = 1 + d + d^2 + \cdots + d^k = \frac{d^{k+1} - 1}{d - 1} \qquad (d > 1) \quad (2)$$

nodes. The graphs achieving this upper bound are called (d, k) *directed Moore graphs* (DMG's). In [15] it is shown that there are no (d, k) DMG's for nontrivial values of d and k ($d > 1$ and $k > 1$).

B. General Graph Construction Techniques

There are two kinds of closely related problems in the design of networks. The first is to maximize the number of nodes n in a graph of degree d and diameter k. We have the upper bounds (1) and (2) and very few graphs, the Moore graphs, achieve them. General techniques for the construction of regular undirected (d, k) graphs with a large number of nodes were given in [6], [1], [7], [10], [14], and [15] for (d, k) directed graphs. However, the gap between the number of nodes achieved by these techniques and the Moore upper bounds is still considerable. For example, the (3, 8) undirected graph constructed by Friedman's technique has 90 nodes when the corresponding Moore bound is $N_{um}(3, 8) = 776$.

The second problem is to minimize the diameter k of a regular graph with n nodes and degree d. This is discussed in [2] and is principally the object of this paper. In fact, this problem is easily generalized to nonhomogeneous graphs. In this case, either the degree of each node or the total number of edges can be given instead of d.

Another closely related generalization we shall discuss is the minimization of the average distance A of a regular graph with n nodes and of degree d where A is defined as

$$A = \sum_{i,j \in V} d_{ij}/n^2.$$

It is easy to compute a lower bound for the average distance A of regular graphs with n nodes and degree d. The method is best explained with an example. Fig. 1 illustrates the breadth-first-search tree of a directed graph with 48 nodes and degree-2 achieving the minimal distance from the root to all other nodes. In this example, the minimal average distance from the root to the other nodes is

$$A = (1 \times 0 + 2 \times 1 + 4 \times 2 + 8 \times 3 + 16 \times 4 + 17 \times 5)/48 = 3.8125.$$

No regular directed graph with 48 nodes and degree-2 can have a smaller value A since this value serves as a bound for all nodes.

II. The Local Search Algorithm

The method of local search has been applied successfully to a number of combinatorial optimization problems, starting with the early work of Bock [4], Lin [11], and Reiter and Sherman [12]. Perhaps the problem closest to the one considered here that has been attacked by this method is the design of low cost networks with prescribed connectivity [13]. A review of the area can be found in [9].

The algorithm can be characterized in general terms as follows. For a given graph G, a neighborhood $N(G)$ of graphs is defined which represents a set of perturbations of the given graph. If a graph $G' \in N(G)$ is found which is an "improvement" of G, it replaces G and the process continues. When finally a graph is obtained which is better than any in its neighborhood, it is accepted as a local optimum with respect to the neighborhood N. We may start the algorithm from several different graphs and accept the best result.

The main choices in the design of such an algorithm are the method of choosing an initial graph, the neighborhood N, the method of enumerating N, and the criterion of improvement. We next take the choices up one at a time for the present problem.

A. The Initial Graphs

In the undirected case, for any given number of nodes n and any degree d, we use the graphs studied by Trufanov in [16] as initial graphs. (Note that n and d cannot be both odd.) Trufanov showed that their diameter is bounded by

$$k \le 1 + \left\lceil \frac{\lceil 2\lfloor m/2 \rfloor / d \rceil + 1}{2} \right\rceil$$

for an even d or

$$k \le 1 + \left\lceil \frac{\lceil n/(d-1) \rceil}{2} \right\rceil$$

for an odd d.

In the directed case, we used a simple variation of Trufanov's graphs. Their diameter is of the same order of magnitude as the diameter of the corresponding undirected graphs.

B. The Neighborhood N

Given a graph $G = (V, E)$, the neighborhood $N(G)$ of G is the set of all graphs G' such that G' is derived from G by an "X-change" [13]. For an undirected graph, an X-change operation consists of replacing any two undirected edges (u, v) and (r, s) such that

$$(u, v) \in E, \qquad (r, s) \in E \qquad (3)$$

$$(u, s) \notin E, \qquad (r, v) \notin E \qquad (4)$$

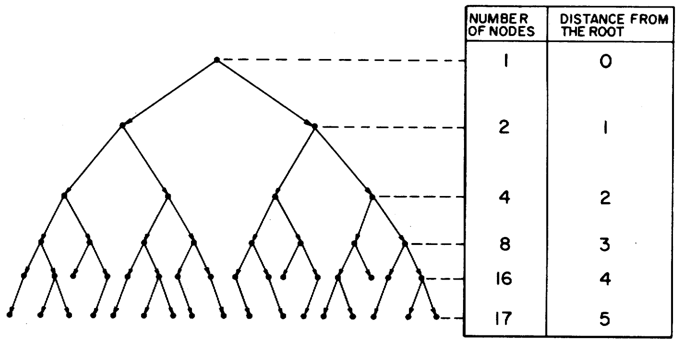

Fig. 1. Example of breadth-first-search tree of a directed graph.

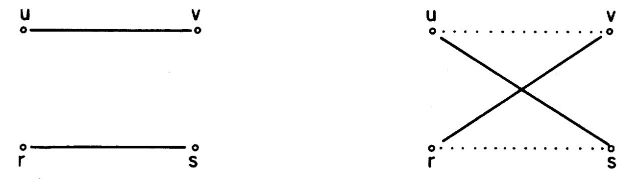

Fig. 2. Undirected X-change operation.

with the edges (u, s) and (r, v). This is illustrated in Fig. 2. For a directed graph, an X-change operation similarly consists of replacing any directed edges (u, v) and (r, s) satisfying condition (3) and (4) with the directed edges (u, s) and (r, s). (Note that the degree, and in the directed case both the in- and out-degree, of each node is not changed by an X-change.)

C. The Method of Enumerating N

Let the (u, v, r, s)-tuple denote the X-change involving the edges (u, v) and (r, s) of a graph $G = (V, E)$. We enumerate the neighborhood $N(G)$ of a graph G by enumerating in lexicographical order all the possible (u, v, r, s) X-changes and then applying them to G.

D. The Criterion for Improvement

In a graph $G = (V, E)$ with diameter k, a pair of nodes $i, j \in V$ is said to be an *extremal pair* if the distance d_{ij} between these two nodes in G is equal to the diameter k of G.

Most of the experimental results shown here were obtained using the following criterion for improvement. Let G be a graph and p and p' denote the respective number of extremal pairs in each graph. G' is an improvement over G if the following condition holds:

$$(k' < k) \vee [(k' = k) \wedge (p' < p)] \tag{5}$$

(G' has a smaller diameter than G or, if it has the same diameter, then it has fewer extremal pairs).

We also tried the following criterion for improvement involving the average distance A. Let $G = (V, E)$ be a graph and let $G' \in N(G)$. Then G' is an improvement over G if the average distance A' in G' is smaller than the average distance A in G. That is if condition

$$A' = \left(\sum_{i,j \in V} d'_{ij}/n^2 \right) < A = \left(\sum_{i,j \in V} d_{ij}/n^2 \right) \tag{6}$$

holds.

III. Experimental Results

Unless otherwise indicated, all of the following discussion refers to results obtained using condition (5) as the improvement condition.

Experimental results show that after a fast initial phase of large improvements, the algorithm spends most of its time in the last few marginal improvements. A typical result is illustrated in Fig. 3. The average distance achieved by the algorithm for a directed graph with 48 nodes and degree-2 is plotted versus the number of X-changes tried up to that point.

For regular graphs with n nodes and degree d, improvement conditions (5) and (6) can be checked in $O(n^2d)$-time. This is done by executing an $O(nd)$-time breadth-first-search about each node of the graph, branching out if and when the improvement condition is verified to be false. We found that most of the algorithm running time is spent in checking the improvement condition.

All the results concern the construction of regular graphs with 8, 16, 24, 32, 40, and 48 nodes. The results for directed graphs are given in Table I and in Table II. In Table I the graphs are of degree-2, in Table II they are of degree-3. Similar results for undirected graphs of degree-3 and degree-4 are given in Table III and Table IV.

We also used condition (6) (minimization of the average distance A) as improvement criterion to construct directed graphs of degree-2. The results are shown in Table V. Comparing this table with Table I it seems that average distance criterion gives consistently better graphs than the previous one. But the improvements were negligible compared to a doubling of the algorithm running time.

Reasonable confidence in the heuristic of this algorithm may be deduced from the following facts.

1) When we tried different initial solutions, they always resulted in local optima with the same diameter and similar (if not equal) number of extremal pairs and average distance. Even using a different minimization criterion we obtained similar results (compare Table I with Table V).

2) A certain consistency of the algorithm behavior is illustrated in Fig. 4 where the number of nodes is plotted versus the average distance of the locally optimal graphs found. A semilogarithmic plot is used, and in all the four cases (directed graphs of degree-2 and degree-3 and undirected graphs of degree-3 and degree-4) the function is almost linear. Therefore, it seems that the number of nodes the algorithm can fit in a regular graph of fixed degree d is an exponential function of the average distance in the graph.

3) We also tried the algorithm for some values of number of

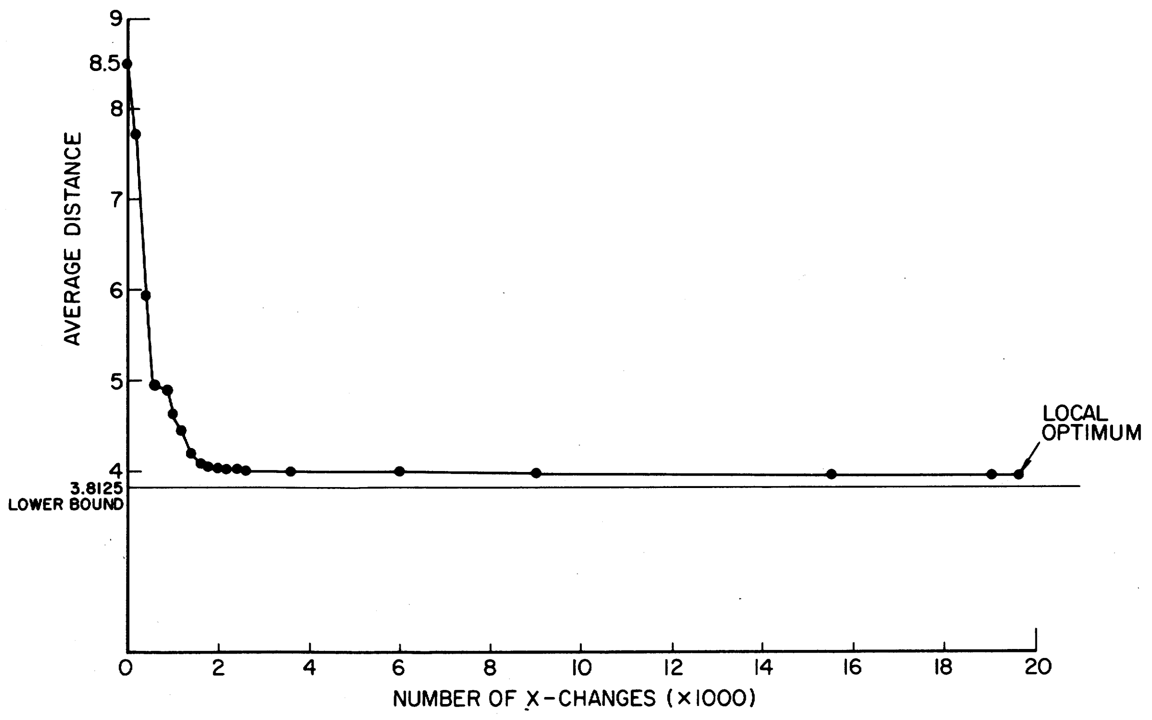

Fig. 3. Average distance versus number of X-changes for the directed case, number of nodes $N = 48$, degree-2.

TABLE I
LOCALLY OPTIMAL DIRECTED GRAPHS OF DEGREE-2

N	d	p	A
8	3	10	1.6562
16	4	42	2.4805
24	5	22	2.9757
32	6	2	3.4169
40	6	43	3.6944
48	6	188	3.9444

N=number of nodes in the graph,
d=diameter of the graph,
p=number of extremal pairs of nodes,
A=average shortest-path distance.

TABLE III
LOCALLY OPTIMAL UNDIRECTED GRAPHS OF DEGREE-3

N	d	p	A
8	2	32	1.3750
16	3	96	2.0625
24	4	48	2.4583
32	5	2	2.8867
40	5	24	3.1037
48	5	254	3.3394

TABLE II
LOCALLY OPTIMAL DIRECTED GRAPHS OF DEGREE-3

N	d	p	A
8	2	32	1.3750
16	3	55	1.9023
24	4	3	2.2829
32	4	55	2.4932
40	4	259	2.7137
48	4	602	2.8867

TABLE IV
LOCALLY OPTIMAL UNDIRECTED GRAPHS OF DEGREE-4

N	d	p	A
8	2	24	1.250
16	3	4	1.6406
24	3	168	2.0416
32	3	480	2.2812
40	4	26	2.4412
48	4	176	2.5894

TABLE V
DIRECTED GRAPHS OF DEGREE 2 OBTAINED BY MINIMIZATION OF THE AVERAGE SHORTEST PATH DISTANCE

N	d	p	A	L	E	n_1	n_2
8	3	10	1.6562	1.6250	1.9	106	9
16	4	40	2.4687	2.3750	3.9	1255	50
24	5	15	2.9531	2.9166	1.25	3829	74
32	6	2	3.3486	3.2187	4.0	6541	130
40	6	33	3.6606	3.5750	2.3	20104	210
48	6	138	3.9079	3.8125	2.5	18247	198

L=a lower bound on the average distance A,
E=100x (A-L)/L,
n_1=number of X-changes tried,
n_2=number of successful X-changes.

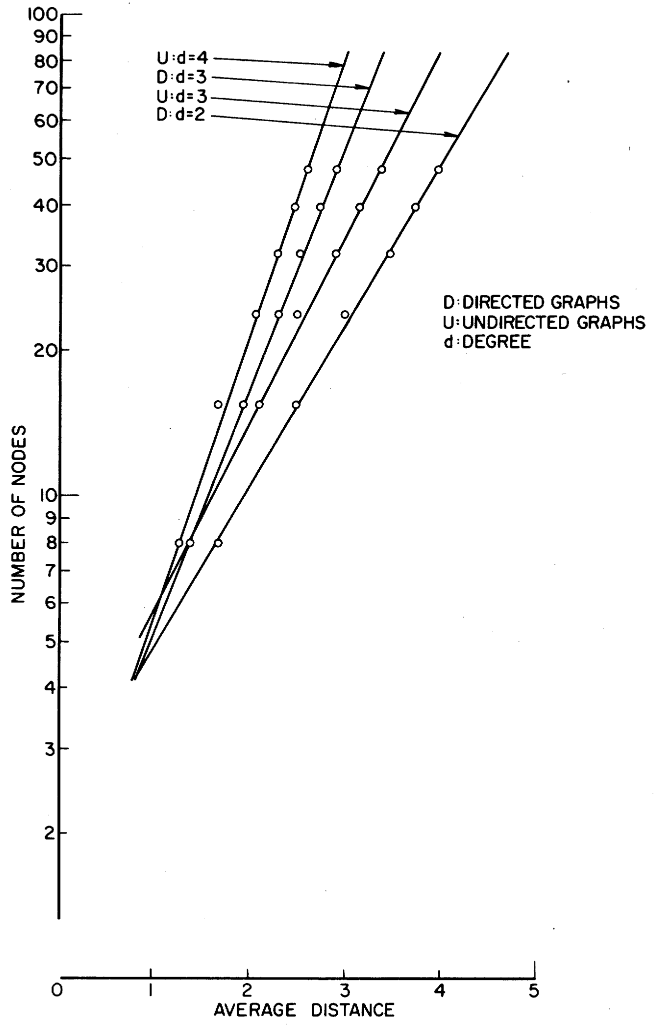

Fig. 4. Number of nodes N versus average distance A.

nodes n and degree d for which maximal undirected graphs (or UMG's) are known [6]. For $n = 10$ and $d = 3$ the local optimum obtained was Petersen's graph, this is the (3, 2) UMG. For $n = 15$ and $d = 4$, the local optimum has the diameter $k = 2$; for $n = 20$ and $d = 3$, the graph obtained has the diameter $k = 3$. Both graphs are known to be maximal and the latter one is illustrated in [6].

4) The average distances obtained are quite close to the known lower bounds. For example, we previously noted that a directed graph with 48 nodes and of degree-2 has a minimum average distance of 3.8125. It is not known if any graph achieves this bound, but the locally optimal graph found, using improvement condition (6), has an average distance equal to 3.9079, which is 2.5 percent more than the lower bound.

TABLE VI
Some Results by Arden and Lee [2]. (Values Obtained by the Local Search Algorithm Are in Parentheses)

	MTS - UNDIRECTED GRAPHS DEGREE = 3			UNDIRECTED GRAPHS* DEGREE = 4		DIRECTED GRAPHS* DEGREE = 2	
N	d	p	A	d	A	d	A
24	4(4)	72(48)	2.500(2.458)	4(3)	2.19(2.04)	6(5)	3.14(2.95)
32	5(5)	32(2)	2.898(2.886)	4(3)	2.38(2.28)	6(6)	3.52(3.35)
40	5(5)	180(24)	3.188(3.104)	4(4)	2.58(2.44)	7(6)	3.82(3.66)
48	5(5)	474(254)	3.453(3.339)	4(4)	2.75(2.59)	7(6)	4.06(3.91)

(*: These graphs are obtained by slight variations of MTS graphs)

5) Finally, the results obtained compare favorably with those known to date. Table VI gives some recent results by B. Arden and H. Lee [2]. It should be pointed out, however, that the graphs derived by Arden and Lee have a known regular structure which can be of great advantage in routing.

This algorithm has two main limitations. One is its impracticability for large values of the number of nodes n ($n > 150$), and the other disadvantage is the lack of knowledge of the structure and of the possible symmetries of the locally optimal graphs obtained. Removing these barriers remains a goal for future work.

References

[1] S. B. Akers, "On the construction of (*d*, *k*) graphs," *IEEE Trans. Electron. Comput.*, vol. EC-14, p. 488, June 1965.
[2] B. Arden and H. Lee, "A multi-tree structured network," submitted for publication.
[3] E. Bannai and T. Ito, "On finite Moore graphs," *J. Fac. Sci., Univ. Tokyo*, vol. 20, pp. 191-208, 1973.
[4] F. Bock, "An algorithm for solving traveling salesman and related network optimization problems," presented at the 14th Nat. Meeting Oper. Res. Amer., St. Louis, MO, Oct. 24, 1958.
[5] R. M. Damerell, "On Moore graphs," in *Proc. Cambridge Phil. Soc.*, vol. 74, 1973, pp. 227-236, 1973.
[6] B. Elspas, "Topological constraints on interconnection limited logic," *Switching Circuit Theory Logic. Des.*, vol. S-164, pp. 133-147, 1964.
[7] H. Friedman, "A design for (*d*, *k*) graphs," *IEEE Trans. Electron. Comput.*, vol. EC-15, pp. 253-254, Apr. 1966.
[8] A. J. Hoffman and R. R. Singleton, "On Moore graphs with diameter 2 and 3," *IBM J. Res. Develop.*, vol. 4, pp. 497-504, 1960.
[9] W. H. Kohler and K. Steiglitz, "Enumerative and iterative computational approaches," in *Computer and Job-Shop Scheduling Theory*, E. G. Coffman, Jr., Ed. New York: Wiley, 1976, ch. 6.
[10] I. Korn, "On (*d*, *k*) graphs," *IEEE Trans. Electron. Comput.*, vol. EC-16, p. 90, Feb. 1967.
[11] S. Lin, "Computer solutions to the traveling salesman problem," *Bell Syst. Tech. J.*, vol. 44, pp. 2245-2269, 1965.
[12] S. Reiter and G. Sherman, "Discrete optimizing," *SIAM J. Appl. Math.*, vol. 13, pp. 864-889, 1965.
[13] K. Steiglitz, P. Wiener, and D. J. Kleitman, "The design of minimum-cost survivable networks," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 455-460, Nov. 1969.
[14] R. M. Storwick, "Improved construction techniques for (*d*, *k*) graphs," *IEEE Trans. Comput.*, vol. C-19, pp. 1214-1216, Dec. 1970.
[15] S. Toueg, "On the impossibility of directed Moore graphs," to be published in the *J. Combinatorial Theory*.
[16] S. V. Trufanov, "Some problems of distance on a graph," *Eng. Cybern.*, pp. 60-66, Jan./Feb. 1967.

Comments on "Generalization of Consensus Theory and Application to the Minimization of Boolean Functions"

ROBERT B. CUTLER and SABURO MUROGA

Abstract—**An algorithm for finding a minimal irredundant disjunctive form representation for an incompletely specified multiple-output switching function presented by P. Tison[1] is shown to be incorrect by counterexample and corrected.**

Index Terms—**Boolean algebra, consensus theory, covering problem, minimal sum, multiple-output functions, prime implicants.**

Step 2 of Algorithm 5 in the paper by P. Tison[1] reads

"2) We keep only the prime implicants of a product $B(F_i) \cdot B(F_k) \ldots$ if the intersection with the product $b(F_i) \cdot b(F_k) \ldots$ is nonempty."

This step will be shown to be incorrect by the following example.

Let a two-output switching function be given by the lower and upper bounds

$$\left\{\begin{aligned} b(F_1) &= ab' + acd \\ B(F_1) &= ab' + ac \\ b(F_2) &= a'b + bcd' \\ B(F_2) &= a'b + bc \end{aligned}\right\}.$$

The upper bounds and products of upper bounds have prime implicants

$$B(F_1)\colon \{ab', ac\}$$
$$B(F_2)\colon \{a'b, bc\}$$
$$B(F_1) \cdot B(F_2)\colon \{abc\}.$$

Manuscript received April 27, 1978. This research was supported in part by the NSF under Grant MCS77-09744.

The authors are with the Department of Computer Science, University of Illinois, Urbana, IL 61801.

[1] P. Tison, *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 446-456, Aug. 1967.