Discussion Paper No. 449

POLYNOMIAL ALGORITHMS FOR ESTIMATING NETWORK RELIABILITY

by

Eitan Zemel*

November 1980

* Supported in part, by NSF Grant ENG-7902896 and by the J. L. Kellogg Center for Advanced Studies in Managerial Economics.

Abstract

We consider the problem of calculating the best possible bounds on the reliability of a system given limited information about the joint density function of its components. We show that a polynominal algorithm for this problem exists iff such an algorithm exists for a certain related problem of minimizing a linear objective function over a clutter. We give numerous examples of network as well as other problems for which the algorithm runs in polynominal time. We also use our construction ro prove NP hardness for others.

Consider a system G which contains a set of n components, $e_1 \dots, e_n$. Let $x_1 \dots, x_n$ be 0-1 random variables which represent the on-off status of these components, and let $X = (x_i)$, $i=1 \dots, n$. We refer to X as the state of the system G. Denote by F(X) the joint probability distribution over the possible states of the system.

For each particular state vector X, the system as a whole may be either on or off. Let $\Phi(\cdot)$ be a function whose domain is the set of possible state vectors and whose range is the set $\{0,1\}$. We say that the system G is "on" under state X iff $\Phi(X) = 1$. The function Φ is often referred to as the structure function of G. In most practical application Φ satisfies the following properties:

(1) $\Phi(0,\dots,0) = 0$

(2) $\Phi(1,\dots 1) = 1$

(3) $Y \geqslant X \Longrightarrow \Phi(Y) \geqslant \Phi(X)$

we restrict our attention in this paper to structure functions Φ satisfying (1) - (3).

Given a structure function Φ and a probability density function F, we can define the reliability of the system

$$r = \text{Prob } [\Phi(X) = 1]$$

For most practical structure functions the calculation of r is extremely tedious even if F itself is simple (e.g. under the assumption that the random variables x_i are mutually independent). For example, let G correspond to an undirected network and let the components be the edges. Let s and t be two distinct nodes of G and consider the structure function

$$\phi(X) = 1 \text{ iff } X \text{ contains a path between s and t}$$

It is known that the problem of calculating r is NP-hard , [14].

In this paper we analyze the problem of computing r when the function F is not completely specified. In particular, we assume that the only information available on this function is in the form of the individual bounds:

(4) $$a_i \leqslant p_i = \text{Prob}\ [x_i = 1] \leqslant b_i \quad i = 1\ldots,n$$

for a given set of constants, $0 \leqslant a_i \leqslant b_i \leqslant 1$, $i=1\ldots,n$. It is apparent that the relations (4) do not, in general, completely specify the function F. (Note that we are <u>not</u> assuming independance of the x's). Consequently, the reliability r is not well defined. What we seek, then, is the best that can be hoped for under the circumstances namely to calculate the best possible upper and lower bounds on r which are consistant with the relations (4). We denote these pounds by β and α respectively. As will be shortly revealed, the calculation of these bounds may in some cases be relatively easy and could be accomplished in polynominal time. In other cases, however, the task of caluculating α or β may turn out to be NP hard.

II. <u>The Algorithm</u>

It will be convenient to describe state vectors as subsets of $N = \{1\ldots,n\}$. For a state vector X let

$$S_X = \{j\varepsilon N : x_j = 1\}$$

and conversely, for a set $S \subseteq N$ let

$$X^S = (x_i^S)\ i\varepsilon N, \text{ with } x_i^S = 1 \text{ iff } i\varepsilon S$$

Let

$$F = \{S \subseteq N : \Phi(X^S) = 1\}$$

then r can be alternatively expressed

$$(5) \qquad r = \text{Prob}\left(\bigcup_{S\varepsilon F} [X = X^S] \right)$$

We note that properties (1)-(3) of the structure function Φ translate into the following properties of the family F:

(1') $\emptyset \notin F$

(2') $N \varepsilon F$

(3') $S \varepsilon F \Longrightarrow S' \varepsilon F$ for every $S' \supseteq S$.

The problem of estimating expression of the type (5) given conditions of the type (4) is old, and was first discussed by Boole [1]. Hailperin, [8], has shown that the best upper bound for such an expresson can be obtained by solving the linear program

$$\beta = \max \sum_{S\varepsilon F} y_S$$

subject to

$$(6) \qquad \begin{aligned} &\sum_{S:i\varepsilon S} y_S \leqslant b_i \quad i\varepsilon N \\ &\sum_{S:i\varepsilon S} y_S \geqslant a_i \quad i\varepsilon N \\ &\sum_{S \subseteq N} y_S \leqslant 1 \\ \\ &y_S \geqslant 0, \quad S \subseteq N \end{aligned}$$

Similarly, the best lower bound on r, α, can be found by solving the minimization problem over the same set of constraints and using the same objective function. We note that the linear program in question involves an exponential number of variables. Thus, any method which relies on explicitly writing down this set of constraints cannot yield a polynominal algorithm for this problem. Nevertheless, the problem contains enough structure to allow implicit handeling of the set of variables. Such an approach gives rise to a polynominal algorithm for α and β for various interesting functions Φ. In other cases, the technique yields an easy proof for the NP hardness of the problem.

The algorithm relies heavily on the recent work of Gröetschel, Lovász and Schrijver [7] (see also Padberg and Rao, [11],) which in turn is a generalization of the polynominal algorithm for linear programming put forward by Katchian, [9].

We recall the essentials of Katchian's method. At the k^{th} iteration we have an elipsoid, E_k, which is known to contain an optimal solution to our problem. Denote the center of this elipsoid by x_k. Assume that x_k is not feasible to our problem. Then any constraint which is violated by x_k can be used to split E_k into two halves such that the optimal solution (which is known to be within E_k) lies, in fact, in one of these halves. Alternatively, if x_k is feasible to our problem, we can achieve a similar halving of E_k using the objective row. The itteration is concluded by enclosing the half elipsoid we wish to keep by another elipsoid, E_{k+1}, whose volume is smaller than that of E_k. One can show that in a plynominal number of steps we obtain an elipsoid whose center is "close" enough to the optimal solution. The procedure is terminated by a process of "rounding" the entries of this center, using, say, continued fractions.

Katchian's algorithm, in the form stated, relies on the possibility to check, in polynominal time, whether or not x_k is feasible. In the negative case we also must be able to specify a violated constraint, i.e. to separate x_k from the feasible set. This relation, between the optimization problem and the "separation problem", is the central theme in Gröetschel, Lovász and Schrijver's paper. The following theorem, which suffices for the purposes of our discussion here, is a consequence of the more general Theorems of [7].

Theorem 1: [7]

Consider the two problems

(P1) max cx

subject to

$$Ax \leqslant b$$

(P2) Given $y \varepsilon R^n$
either show $Ay \leqslant b$
or find a violated constraint $a_i y > b$.

Assume that the entries in c,A and b are integers whose magnitude is bounded by T. Assume further that the feasible set of (P1) is full dimensional and that a point a_0 exists in the interior of this set such that each entry of a_0 can be expressed as the ratio of two integers whose magnitude is bounded by T. Finally, let us restrict our attention to vectors y satisfying the same property as a_0. Then there exists an algorithm for (P1) which is polynomial in n and log T iff such algorithm exists for (P2).

In order to apply Theorem 1 to the linear programming program (6) we need to slightly modify this problem by taking its dual and eliminating reduntant constraints.

Let

$$H = \{\bar{S}: S \notin F\}$$

where $\bar{S}$ represents the complement of S in N. Let F^* and H^* be the set of minimal elements in F and H respectively. Then both G^* and F^* are clutters, i.e. each is a collection of subsets none of which contains the other. Furthermore F^* and H^* constitute a blocking pair, [5], i.e. H^* can be defined via the relation

$$H^* = \{S: X^S X^T \geq 1 \text{ for every } T \varepsilon F^* \text{ and S is minimal with respect to this property}\}$$

Coversely, F^* can be defined from H^* using the same relation. Finally, let $\bar{a}_i$ denote the complementary probability $1-a_i$.

Theorem 2 Let Φ satisfy the properties (1) - (3). Then,

$$\beta = \min \sum_{i=1}^{n} u_i b_i + w$$

(a) subject to $\sum_{i\varepsilon S} u_i + w \geq 1$ for every $S \varepsilon F^*$

$$u, w \geq 0$$

$$\alpha = 1 - \min \left(\sum_{i=1}^{n} u_i \bar{a}_i + w \right)$$

(b) subject to

$$\sum_{i\varepsilon S} u_i + w \geq \text{ for every } S\varepsilon H^*$$

$$u, w \geq 0$$

Proof

(a) Take the linear programming dual of (6)

$$\beta = \min \sum_{i=1}^{n} u_i b_i - \sum_{i=1}^{n} v_i a_i + w$$

subject to

(7)

$$\sum_{i\varepsilon S} u_i - \sum_{i\varepsilon S} v_i + w \geq 1 : S\varepsilon F$$

$$\sum_{i\varepsilon S} u_i - \sum_{i\varepsilon S} v_i + w \geq 0 : S\varepsilon F$$

$$u, v, w \geq 0$$

We first note that there exists an optimal solution to (7) where for each pair u_i, v_i, at most one member is positive. For otherwise we can decrease both u_i and v_i by $\min (u_i, v_i)$. This will leave the solution feasible without increasing the objective value (since $b_i \geq a_i$). Next we prove that there exists an optimal solution with $v_i = 0$, $i=1\ldots n$. Let (u,v,w) be an optimal solution which does not satisfy this property but for which $v_i \cdot u_i = 0$, $i = 1\ldots,n$. . Let $N_1 \_ N$ be the set of indices for which $v_i > 0$. Let $\bar{v} = \sum_{i\varepsilon N_1} v_i$. Consider the inequality which correspond to N_1.

Irrespective of whether $N_1 \varepsilon F$ or not, we can conclude

$$\sum_{i\varepsilon N_1} -v_i + w \geqslant 0$$

i.e. $\bar{v} \leqslant w$. Consider the point $(u,0,w-\bar{v})$. Since $a_i \leqslant b_i$, $i=1...n$, the objective function of this point is not worse than that of (u,v,w). Next, we demonstrate that this point is feasible for (7).

Assume, in the negative, that the constraint associated with some set S is violated. Without loss of generality we can assume that $S \supseteq N_1$ since otherwise we can add to S the missing elements of N_1 without changing the left hand side of the constraints but with possible increase in the right hand side. But then, for S which contains N_1, the left hand side calculated with respect to (u,v,w) is the same as the left hand side calculated with respect to our new point $(u,0,w-\bar{v})$. Thus, we have contradicted the assumption that the constraint associated with S is violated. The proof is completed by noting that the constraints which correspond to sets $S \notin F$, and to sets S in F which are not minimal there, are redundant.

(b) The expression for α follows from that of β by considering the complementary events $\Phi(X) = 0$ and $x_i = 0$, $i = 1...n$.

For a vector $x \varepsilon R^n$ of rational numbers, let $T(x)$ be the smallest integer such that each entry of x can be expressed as the ratio of two integers bounded in magnitude by $T(x)$. Let $Q(x)$ be the smallest integer such that $Q(x)\ x$ has all components integral. Obviously,

$$T(x) \leqslant Q(x) \leqslant T(x)^n$$

Theorem 3. Let Φ satisfy properties (1) - (3).

(a) There exists a algorithm for calculating β which is polynomial in (n,log T(b)) iff there exists an algorithm for the problem

$$\min_{S \varepsilon F^*} c\, X^S$$

which is polynomial in (n,log (T(c)) for every non negative vector c.

(b) Same as (a) with α replacing β, H* replacing F*, and log $T(\bar{a})$ replacing log T(b).

Proof.

(a) We first consider the polyhedron of Theorem 2. Obviously this polyhedron is full dimensional and contains an interior point as requested (e.g. $u_i=2$ $i=1...n$, $w=2$). Furthermore, its constraints matrix is made out of zeros and ones only. Thus, it satisfies the stipulations of Theorem 1 for any $T \geqslant T(b)$. It follows that there exists an algorithm for β which is polynomial in (n, log T) iff for every vector u with $T(u) \leq T$ and for any rational w which can be expressed as the ratio of two integers bounded by T, we can decide in polynomial time whether or not (u,w) satisfies all the constraints of this polyhedron.

Assume that a polynomial algorithm for min $cX^S : S \varepsilon F^*$ does exist. Then running this algorithm with u replacing c we can obviously decide whether (u,w) satisfies all the constraints and in the negative case find a violated constraint (namely the one which yields the minimum in the minimization problem). Thus, we have demonstrated the existance of an algorithm for β which is polynomial in (n,log T(b)). Conversely, suppose that such an algorithm for β exists. Then we know by theorem 1 that a polynomial algorithm exists for deciding, for each w and cost vector c, whether or not $\min cX^S : S \varepsilon F^*$ is greater than a equal to w. From the stipulations on the vector c it follows that the optimal value of this problem, w*, is a rational satisfying

$$w^* \varepsilon [0, \sum_{i=1}^{n} c_i]$$

and furthermore, $w^* = p/q$, with, $p \geq 0$, $q > 0$, $\max \{p,q\} \leqslant Q(c) \leqslant T(c)^n$. Thus, w^* can be found in $O(\log(T(c)^n) = O(n \log(T(c))$ applications of this algorithm, [12], [13], i.e. in polynomial time as asserted.

Before examining specific applications of Theorem 3, we consider a corollary of this Theorem. Let $\Phi_1 \ldots . \Phi_k$ be a given set of structure functions defined on the same system G. For i = 1...k. Let α_i be the lower bound on r defined with respect to Φ_i. Similarly define β_i, F_i, F_i^*, and H_i^*, i = 1...k.

<u>Correlary 3.1.</u> (a) Assume there exist polynomial (in n, log T(b)) algorithms for calculating β_i, i = 1...k. Then, there exist a polynomial (in (n,k, log T(b)) algorithm for calculating β for the structure function Φ satisfying

$$\Phi(X) = \max_{i=1..k} \Phi_i(X)$$

(b) Assume there exist polynomial algorithms for calculating

α_i, i = 1...k. Then there exists a polynomial algorithm for calculating α for the function

$$\Phi(X) = \min_{i=1...k} \Phi_i(X).$$

Proof.

We note that $\Phi(X) = \max_{i=1...k} \Phi_i(X)$

implies $F = \bigcup_{i=1...k} F_i$ and $F^* \subseteq \bigcup_{i=1...k} F_i^*$. Thus, we can minimize cx

over F* by minimizing over each of the F^*_i's and then taking the grand minimum. Similarly, $\Phi(x) = \min_{i=1...k} \Phi_i(x)$ implies that $H^* \subseteq \bigcup_{i=1...k} H_i^*$.

We now consider some specific structure functions Φ and examine them vis-a-vis the stipulations of Theorem 3. In cases (a) - (h) below, the random variables x_i, i=1...n correspond to the on-off status of the edges of the graph in question.

(a) Let G be a directed graph, s and t two specific nodes. Assume that G is considered "on" iff the graph induced by state vector X is such that there exists at least one directed path from s to t. We note that

F^* = { set of all s-t pathes in G}

H^* = { set of all s-t cuts in G}

since a polynomial algorithm exists for minimizing a non-negative linear function both on F^*, and on H^*, we can conclude that there exists a polynomial algorithm for α and for β for this case.

(b) Consider a graph as in (a) but with undirected edges. Again we let $\Phi(X) = 1$ iff the graph induced by X contains an (undirected) path between two specific nodes, s and t. Note that F^* and H^* in this case are the undirected versions of the sets F^* and H^* discussed in (a). It is well known that the optimization problems over both H^* and F^* can be solved in polynomial time. Thus, we have a polynomial algorithm for α and β in this case too.

(c) Let G be a directed network, s a specific node. Let $\Phi(X) = 1$ iff the graph induced by X contains a directed path from s to every other node of G. We note that F^* is the set of arborescences of G, rooted at s. Thus, there exists a polynomial algorithm for calculating β, [4]. Also, by applying corollary 3.1 to case (a) considered earlier, we know that there exists a polynomial algorithm for calculating α for this case

(d) Let G be an undirected network, and assume that G is "on" iff in the graph induced by X there exists a path connecting every pair of nodes of G. There exists a polynomial algorithm for calculating β since F^* in this case is the set of spanning trees of G. A polynomial algorithm for α can be obtained by applying corollary 3.1 to the problem discussed in part (b).

(e) To contrast the four cases considered earlier, consider again an undirected network G and assume that G is "on" iff the graph induced by X contains an Hamiltonian tour of G. It follows that F^* is the set of Hamiltonian tours in G and H^* is the blocker clutter of this set. Note that the problem of deciding whether F^* is empty for a given graph is NP complete. Thus, the optimization problem over either F^* or H^* is NP hard and so are, then, the problems of calculating a and b.

(f) (Network Reliability, [6]): To note the distinction between calculating a and b we consider an undirected Network G. Let $v_1 \ldots v_k$ be a specific subset of the nodes of G. Consider the structure function

$D(X) = 1$ iff the graph induced by X contains a path between every pair of nodes v_i, v_j $1 F i < j F k$.

Then F^* is the set of Steiner trees defined on G with respect to node set $v_1, \ldots v_k$. It is well known, [14], that minimization of a linear function over this clutter in NP complete. On the other hand, the problem of calculating a can be accomplished in polynomial time by applying corollary 3.1 to the problem discussed in part (b).

(g) We note a certain converse to the problem discussed in (f). Consider again an undirected graph G with a specified set of nodes $v_1, \ldots v_k$. Let $D(X) = 1$ iff the graph induced by X contains a path between <u>at least</u> one pair of nodes v_i, v_j, $1 F i < j F k$. Then F^* can be described as a union of set of paths in G (between every pair of nodes v_i, v_j, $1 F i < j F k$) and thus one can calculate b in polynomial time. On the other hand H^* can be described as the intersection of sets of cuts on G. The status of the optimization problem over H^* is unknown.

(h) Let G be an undirected graph. We say that G is on iff the graph induced by X contains a perfect matching i.e. we can use the "on" edges of G to pair the nodes of G in such a way that each node is paired to exactly one other node. Obviously, the set F* correspond to the set of perfect matching of G and minimization over this set can be achieved in polynomial time [2]. The status of the minimization problem over the blocker set of F* is unknown. However if G is bipertite, H^* is known, [5], and a polynominal algorithm exists for mininizing over this set. Thus, for such graphs, we can calculate α in polynominal time. The assertions of this part are valid for graphs G which do not contain a perfect matching if we replace "perfect" by "Maximum cordinality" matching in the definition of Φ .

(i) (Network Survivability, [6]): Let G = (V,E) be an undirected graph such that both its edges and its nodes are subject to failure. Let $V \bigcup E = K = \{k_1 \ldots k_n\}$. Then the components of the state vector X correspond to the elements of K. We say that the network <u>survives</u> (i.e. $\Phi(X) = 1$) iff for every edge i ε E, e = (u, v), at least one of the triple e, u, v is "on".

For each edge e = (v,u) ε E, let i(e), j(e), k(e) be indices such that $k_{i(e)} = u$, $k_{j(e)} = v$, $k_{h(e)} = e$ and let X^e be the vector: $x_i^e = 1$ iff i = i(e), j(e) or h(e).

It is easy to verify that

$$H_k^* = \{x^e\}_{e \; \varepsilon \; E}$$

minimizing over H^* is trivial, and so α can be calculated polynomially. On the other hand the set F_i^* correspond to the set of minimal covers (of edges) by edges and nodes. Minimizing a linear function over F_i^* is NP complete since it is a more general problem than the node covering (of edges) problem [10]. Thus, calculating β here is NP hard.

The last two examples considered refered to a general system, not necessarily associated with a network.

(j) Consider a system with components $e_i \ldots e_n$ and let K be a matroid over $N = \{ i \ldots n\}$. Assume that $\Phi(x) = 1$ iff the vector X contains a basis of K. Then

$$F^* = \{\text{set of bases of } k\}$$
$$H^* = \{\text{set of co-circuits of } k\}$$

Minimization over F* can be accomplished in polynomial time using the greedy algorithm [3]. Thus, β can be calculated in polynomial time for this problem. The status of the minimization problem over H* is unknown.

(k) Let $e_i \ldots e_n$ be a set of elements each associated with a value c_j, j=1...n. Consider a system which is "on" iff the value of the elements which are "on" is at least b. Then

$$F = \{S: \sum_{j \in S} c_j \geqslant b\}$$
$$H = \{S: \sum_{j \in S} c_j \geqslant \sum_{j \in N} c_j - b\}$$

with F*, H* being the minimal elements in F and H respectively. Both minimization problems in this case are NP-complete. However, if $c_j = 1$, $j = 1 \ldots n$, and $1 \leq b \leq n$ we get a system which is on iff at least b components out of n are on. Thus

$$F^{*} = \{S: |S| = b\}$$
$$H^{*} = \{S: |S| = n - b\}$$

Both F^{*} and H^{*} are known to correspond to the set of bases of especially simple matroids. Thus, by the reasoning of the previous case, α and β can be solved in polynomial time. An obvious application of this case is the one of estimating the probability of a favorable outcome in a voting system with two possible outcomes where all voters counts equally, and with a given treshhold of acceptance, given that we only know bounds on the individual probability of voting for and against this outcome.

(1) Consider a graph with a node set V. Consider two sets of subsets of V, Σ and Δ. Σ can be thought of as a set of centers located in G. Δ,on the other hand, corresponds to a set of points where demand exists for the services offered by these service centers. Let $\Sigma = \{V_i\}_{i\varepsilon N}$, $\Delta = \{V_j\}_{j\varepsilon K}$ for N and K two index sets, not necessarily distinct. Assume that the supply points in Σ are not reliable and let $X = (x_i)_{i\varepsilon N}$ be the state vector of their on-off status. Assume that the system is "on" iff every demand point in $y_j \varepsilon \Delta$ is at a distance of no more than r_j units from a service center $\nu_i \varepsilon \Sigma$ which is operating i.e such that x_i=1. Let F be the set of subsets of N which satisfy the required property. Minimizing a linear objective over this set is solved in polynomial time in [15]. Thus β can be found in polynomial time. To show that α can be found in polynomial time as well we note that H can be written as

$$H = \bigcup_{j\varepsilon K} H\,(j)$$

$$H^* \subseteq \bigcup_{j \varepsilon K} H^* (j).$$

where H(j) is the set of subsets of N such that N/H(j) contains only elements of N which are at distance of more than r units away from v_j. Thus, $H^*(j)$ contains a unique subset, namely the one which contains all the elements of N which are at distance of r_j or less from v_j. Thus, miminization over H^* is trivial.

REFERENCES

[1] Boole, G. "Studies in Logic and Probabilities", Walts and Co. 1952.

[2] Edmonds, J. "Paths, Trees and Flowers" can. J. Math., 17 (1965) 449-467.

[3] ___________, Matroids and the greedy Algorithm, Math. Prog 1 (1971) 127-136.

[4] ___________, "Optimum Branchings" J. Res NBS 71B (1967) 233-240 and also Mathematics and the Decision Sciences, Part 1, G. B. Dantzig and A. F. Veinott, Jr. (eds.), Amer Math Soc. Lectures Appl. Math. 11 (1968) pp. 333-345.

[5] Fulkerson, D. R. "Blocking Polyhedra" in: Graph Theory and its Applications" (Proc. Adv. Seminar, Madison, Wis., 1969, B. Harris ed.), Acad. Press, New York 1970, pp. 43-112.

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

[7] Grötschel, M., Lovász and A. Schrijver "The Ellipsoid Method and its Consequences In Combinational Optimization" Report No. 80151 - OR, Institut fur Okonometrie und Operations Research, Universitat Bonn, W. Germany, January 1980.

[8] Hailperin, T. "Best Possible Inequalities for the Probabilities of a Logical Function of Events" Amer. Math Monthly, 72 (1965) pp. 343-359.

[9] Katchian, L. G. "A polynomial Algorithum in Linear Programming" English Translation: Soviet Math. Dokl. 20 (1979) pp. 191-194.

[10] Karp. R. M. "Reducability Among Combinational Problems" In R. E. Miller and J. W. Thatcher (eds), Complexity of Computer Computations, (1972) Plenum Press, New York, 85-104.

[11] Padberg, M. and Rao "The Russian Method and Integer Programming", ORSA/TIMS Joint National Meeting Colorado Springs, November 1980.

[12] Papadimitriou, C. H. "Efficient Search for Rationals" Inf. Proces Letters Vol. 8 No. 1 (1979) pp. 1-4.

[13] Reiss, S. P., "Rational Search" Inf. Proces. Letters Vol. 8 No. 2 (1979) pp. 89-90.

[14] Rosenthal, A. "Computing Reliability of Complex Systems" Doctoral Thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA (1974).

[15] Tamir, A. "A class of Balanced Matrices Arising From Location Problems" Department of Statistics, Tel-Aviv University, (1980).