Some Network Flow Problems Solved with Pseudo-Boolean Programming

Author(s): Peter L. Ivănescu
Source: *Operations Research*, Vol. 13, No. 3 (May - Jun., 1965), pp. 388-399
Published by: INFORMS
Stable URL: http://www.jstor.org/stable/167803
Accessed: 14-01-2016 09:48 UTC

SOME NETWORK FLOW PROBLEMS SOLVED WITH PSEUDO-BOOLEAN PROGRAMMING

Peter L. Ivănescu

Institute of Mathematics, Romanian Academy, Bucharest, Romania

(Received December 4, 1963)

The aim of this paper is to apply the method of pseudo-Boolean programming to the determination of the minimal cut and of the value of the maximal flow through a network without, or with, given lower bounds on the arc flows, as well as to the solution of some feasibility problems in networks.

THE AIM of the present paper is to apply the method of pseudo-Boolean programming[11] to the determination of the minimal cut and of the value of the maximal flow through a network, to extend this procedure to the case when lower bounds on the arc flows are given, as well as to apply this method to the determination of the feasibility of some supply-demand problems and of the existence of circuits in a network.

In a paper now in preparation it will be shown that many other problems on networks may be efficiently treated by means of pseudo-Boolean programming.

The same method of pseudo-Boolean programming was applied for finding the minimal number of rows and columns of a matrix covering its zero elements in the Hungarian method of solving transportation problems,[13,14] and to the minimization of Boolean functions,[12] a problem arising in switching algebra. In reference 10 it was shown that the problems of determination of the chromatic number, of the number of internal stability, of the number of external stability, and of the kernels of a finite graph may be all treated by means of pseudo-Boolean programming.

In references 8 and 9 it was shown that all problems of integer mathematical programming, where both the objective function and the restraints are polynomials and where the unknowns have given upper and lower bounds, may be solved with the aid of the same technique.

Throughout this paper we shall use the terminology of reference 3.

By a network $G=[N,A]$ we mean a finite number of elements $s_1, s_2, \cdots, s_n \epsilon N$ and a subset A of ordered pairs (s_i, s_j), with $s_i, s_j \epsilon N$. s_i are called nodes; $(s_i, s_j)\epsilon A$ are called arcs.

The node s_1 is called the source, the node s_n is called the sink of the network.

We define a n by n matrix $\mathbf{A}=\|a_{ij}\|$ by taking

$$a_{ij}=\begin{cases}1 \text{ if } (s_i, s_j)\in A,\\ 0 \text{ if } (s_i, s_j)\notin A.\end{cases}$$

PSEUDO-BOOLEAN PROGRAMMING*

LET L_2 be the Boolean Algebra with two elements 0 and 1, its operations "$\cup$", "$\cdot$", "$^{-}$" being defined by:

$\cup$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

a	0	1
$\bar{a}$	1	0

We notice that

$$a\cup b=a+b-ab, \tag{1}$$

and

$$\bar{a}=1-a,$$

where $+$, $-$, and juxtaposition denote the usual addition, subtraction, and multiplication.

A function

$$F: L_2^n\rightarrow R$$

is called a pseudo-Boolean function; here L_2^n is the Cartesian product

$$\underbrace{L_2\times L_2\times\cdots\times L_2,}_{n}$$

while R is the field of the real numbers. It has been proved in reference 11 that any pseudo-Boolean function may be written as a polynomial with real coefficients.

By a problem of pseudo-Boolean programming we mean the determination of all points $(x_1, \cdots, x_n)\in L_2^n$ minimizing a pseudo-Boolean function $F_1(x_1, \cdots, x_n)$.

The following procedure is given for minimizing a pseudo-Boolean function F_1. Let us put

$$F_1=x_1F_{11}+\bar{x}_1F_{12}+F_{13}=x_1F_{11}+(1-x_1)F_{12}+F_{13}$$

or,

$$F_1=x_1g_1+h_1,$$

where F_{11}, F_{12}, F_{13}, $g_1=F_{11}-F_{12}$, $h_1=F_{12}+F_{13}$ are pseudo-Boolean functions of $(x_2, x_3, \cdots, x_n)$.

Let us put

$$x_i^{\alpha_i}=\begin{cases}x_i & \text{if } \alpha_i=1,\\ \bar{x}_i=1-x_i & \text{if } \alpha_i=0.\end{cases} \tag{2}$$

* For proofs and details, *see* reference 11.

Let us now denote

$$M_1=\{(\alpha_2, \cdots, \alpha_n)\in L_2^{n-1}|g_1(\alpha_2, \cdots, \alpha_n)<0\},$$

$$N_1=\{(\beta_2, \cdots, \beta_n)\in L_2^{n-1}|g_1(\beta_2, \cdots, \beta_n)=0\},$$

and put

$$x_1=\bigcup_{(\alpha_2, \cdots, \alpha_n)\in M_1} x_2^{\alpha_2}\, x_3^{\alpha_3}\cdots x_n^{\alpha_n} \cup u_1[\bigcup_{(\beta_2, \cdots, \beta_n)\in N_1} x_2^{\beta_2}\, x_3^{\beta_3}\cdots x_n^{\beta_n}], \quad (3.1)$$

where u_1 is an arbitrary parameter in L_2 (i.e., it may be equal to 0 or to 1), and where

$$\bigcup_{j\in\{k_1, \cdots, k_p\}} y_j=y_{k_1}\cup y_{k_2}\cup\cdots\cup y_{k_p}$$

is defined through the repeated application of formula (1) [$A\cup B=A+B-AB, A\cup B\cup C=(A\cup B)\cup C=A+B-AB+C-(A+B-AB)C=A+B+C-AB-BC-CA+ABC$, etc.] and where, of course, we consider

$$\bigcup_{j\in\phi} y_j=0.$$

x_1 is expressed as a function of $u_1, x_2, \cdots, x_n$:

$$x_1=x_1(u_1, x_2, \cdots, x_n). \quad (3.1')$$

Let us denote

$$x_1^0=x_1(0, x_2, \cdots, x_n),$$

or, according to (3.1):

$$x_i^0=\bigcup_{(\alpha_2\cdots\alpha_n)\in M_1} x_2^{\alpha_2}\, x_3^{\alpha_3}\cdots x_n^{\alpha_n}. \quad (3.1^0)$$

Through the repeated application of formulas (1) and (2), we see that we can transform the expression (3.1^0) of x_i^0 into another, say x_1^+, which will contain only the usual arithmetical operations $+, -, \cdot$.

Let us denote with $F_2(x_2, \cdots, x_n)$ the pseudo-Boolean function obtained putting in $F_1(x_1, x_2, \cdots, x_n)$,

$$x_1=x_1^+(x_2, \cdots, x_n),$$

i.e.,

$$F_2(x_2, \cdots, x_n)=F_1[x_1^+(x_2, \cdots, x_n), x_2, \cdots, x_n].$$

As F_2 is a pseudo-Boolean function we may write

$$F_2(x_2, \cdots, x_n)=x_2g_2(x_3, \cdots, x_n)+h_2(x_3, \cdots, x_n).$$

Let us put, as above,

$$M_2=\{(\alpha_3, \cdots, \alpha_n)\in L_2^{n-2}|g_2(\alpha_3, \cdots, \alpha_n)<0\},$$

$$N_2=\{(\beta_3, \cdots, \beta_n)\in L_2^{n-2}|g_2(\beta_3, \cdots, \beta_n)=0\},$$

and

$$x_2=\bigcup_{(\alpha_3, \cdots, \alpha_n)\in M_2} x_3^{\alpha_3} x_4^{\alpha_4}\cdots x_n^{\alpha_n} \cup u_2[\bigcup_{(\beta_3, \cdots, \beta_n)\in N_2} x_3^{\beta_3} x_4^{\beta_4}\cdots x_n^{\beta_n}], \quad (3.2)$$

where u_2 is an arbitrary parameter in L_2.

Generally we put

$$x_i = \bigcup_{(\alpha_{i+1},\cdots,\alpha_n)\epsilon M_i} x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n} \cup u_i [\bigcup_{(\beta_{i+1},\cdots,\beta_n)\epsilon N_i} x_{i+1}^{\beta_{i+1}} \cdots x_n^{\beta_n}], \quad (3.\mathrm{i})$$

$$x_n = \begin{cases} 1 & \text{if the constant } g_n < 0, \\ 0 & \text{if the constant } g_n > 0, \\ u_n & \text{if the constant } g_n = 0. \end{cases} \quad (3.\mathrm{n})$$

From (3.n) we see that

$$x_n = X_n(u_n) \quad (4.\mathrm{n})$$

[where $X_n(u_n)$ is equal to 1 or to 0 or to the free parameter $u_n \epsilon L_2$].

From (3.n) and (3.n-1),

$$x_{n-1} = x_{n-1}(u_{n-1}, x_n) = x_{n-1}[u_{n-1}, X_n(u_n)] = X_{n-1}(u_{n-1}, u_n) \quad (4.\mathrm{n}\text{-}1)$$

analogously,

$$x_i = X_i(u_i, u_{i+1}, \cdots, u_n), \quad (4.\mathrm{i})$$

$$\cdots$$

$$x_1 = X_1(u_1, u_2, \cdots, u_n). \quad (4.1)$$

It is proved in reference 11 that for each system of values of the parameters $u_1, u_2, \cdots, u_n$ the system (4.n), (4.n-1), $\cdots$, (4.i), $\cdots$, (4.1) yields a minimum of F_1, and conversely, any minimum of F_1 can be obtained in this way.

NOTE. The computation of $x_1, x_2, \cdots, x_n$ may be carried out in an order different from the above, if this seems to be more convenient. Denoting with N_{ij} the number of terms of F_i containing x_j, it is more efficient to start with the computation of that x_k for which $N_{ik} = \min_j N_{ij}$.

It is easy to notice that, given a set $Q \epsilon L_2{}^m$ with the property that there is an $r(0 < r \leqq m)$ so that

$$(\gamma_1, \cdots, \gamma_r, \gamma_{r+1}, \cdots, \gamma_m) \epsilon Q \Rightarrow (\gamma_1, \cdots, \gamma_r, \gamma_{r+1}', \cdots, \gamma_m') \epsilon Q,$$

then putting

$$Q' = \{(\gamma_1, \cdots, \gamma_r) \epsilon L_2{}^\gamma | \text{there exists } (\gamma_{r+1}, \cdots, \gamma_m) \epsilon L_2^{m-r} \text{ so that } (\gamma_1, \cdots, \gamma_m) \epsilon Q\}$$

we have

$$\bigcup_{(\gamma_1,\cdots,\gamma_m)\epsilon Q} x_1^{\gamma_1} \cdots x_r^{\gamma_r} x_{r+1}^{\gamma_{r+1}} \cdots x_m^{\gamma_m} = \bigcup_{(\gamma_1,\cdots,\gamma_r)\epsilon Q'} x_1^{\gamma_1} \cdots x_r^{\gamma_r}. \quad (5)$$

This type of simplification is frequently used.

Of course if $r = 0$, we have

$$\bigcup_{(\gamma_1,\cdots,\gamma_m)\epsilon Q} x_1^{\gamma_1} \cdots x_m^{\gamma_m} = 1. \quad (5')$$

In the following paragraphs, examples of application of the above procedure of minimizing a pseudo-Boolean function will be given.

We have solved all the numerical examples by hand computation, and the procedure proved to be very efficient. Its coding for a MECIPT-1 computer is in progress.

MAXIMAL FLOWS IN A NETWORK

GIVEN A network $G=[N,A]$ we shall suppose that we are also given a matrix $C=\|c_{ij}\|$ of nonnegative elements, called arc-capacities, satisfying

$$a_{ij}=0\Rightarrow c_{ij}=0. \tag{6}$$

A *flow* of value v from the source s_1 to the sink s_n is a matrix $F=\|f_{ij}\|$ of nonnegative elements called *arc-flows* satisfying the following conditions:

$$\sum_{j=1}^{j=n} a_{ij}f_{ij}-\sum_{j=1}^{j=n} a_{ji}f_{ji}=\begin{cases} v & \text{if } i=1, \\ 0 & \text{if } i\neq 1, n, \\ -v & \text{if } i=n, \end{cases} \tag{7}$$

$$f_{ij}\leqq c_{ij}. \tag{8}$$

The maximal flow problem is that of finding those f_{ij} for which v would be maximal, conditions (7) and (8) being satisfied.

If $M\subseteq N$ we shall put $M'=N-M$ and $I_M=\{i|s_i\epsilon M\}$. For any function

$$g: A\rightarrow R,$$

we put

$$g(M, M')=\sum_{i\epsilon I_M}\sum_{j\epsilon I_{M'}} g_{ij}. \tag{9}$$

By a *cut* C in $[N; A]$ separating s_1 and s_n we mean a set $K\subset N$ so that $s_1\epsilon K$, $s_n\epsilon K'$. The capacity of a cut is $c(K, K')$. The minimal cut problem is that of determining a cut of minimal capacity.

FORD AND FULKERSON[1,2] have proved the following fundamental result in the theory of flows in networks:

For any network the maximal flow value from s_1 to s_n is equal to the minimal cut capacity of all cuts separating s_1 and s_n.

When a minimal cut K_0 is known, then

$$v_{\max}=c(K_0, K_0').$$

Any subset M of N may be characterized by the vector $(x_1, x_2, \cdots, x_n)$ where

$$x_i=\begin{cases} 1 & \text{if } i\epsilon I_M, \\ 0 & \text{if } i\notin I_M. \end{cases}$$

A cut C separating s_1 and s_n will be characterized by a vector $(1, x_2, x_3, \cdots, x_{n-1}, 0)$.

The capacity of a cut is

$$c(K,\,K')=\sum_{i\epsilon I_M}\sum_{j\epsilon I_{M'}} c_{ij}=\sum_{i=1}^{i=n}\sum_{j=1}^{j=n} c_{ij}\,x_i\,\bar{x}_j.$$

As $x_1=1$, $x_n=0$, we have

THEOREM 1. *If* $(1,\,x_2^0,x_3^0,\,\cdots,x_{n-1}^0,0)$ *gives a minimum of the pseudo-Boolean function*

$$G=c_{1n}+\sum_{j=2}^{n-1} c_{1j}\,\bar{x}_j+\sum_{i=2}^{n-1} c_{in}\,x_i+\sum_{i=2}^{n-1}\sum_{j=2}^{n-1} c_{ij}\,x_i\,\bar{x}_j, \quad (10)$$

then

$$K_0=\{s_i|x_i^0=1\}$$

is a minimal cut separating s_1 *and* s_n, *and conversely, any minimal cut separating* s_1 *and* s_n *may be obtained in this way.*

Thus, the problem of finding a minimal cut separating the source and the sink (and implicitly that of determining the value of a maximal flow) in a network is reduced to a problem of pseudo-Boolean programming.

It was pointed out by one of the referees that, from a computational point of view, the method is the more efficient the greater is the number of nodes as compared to that of arcs; in particular when the minimum degree of the nodes is small, then the procedure appears to be rapid. I fully subscribe to his remark.

Example. Let us consider the network discussed in reference 5,

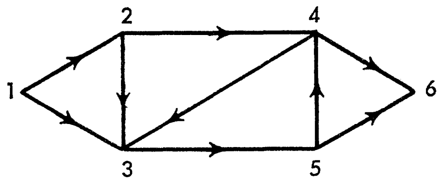

Figure 1

having:

c_{ij}	1	2	3	4	5	6
1	0	7	3	0	0	0
2	0	0	1	6	0	0
3	0	0	0	0	8	0
4	0	0	3	0	0	2
5	0	0	0	2	0	8
6	0	0	0	0	0	0

Putting

$$x_1=1, \quad (11.1)$$

we have

$$G_2=7\bar{x}_2+3\bar{x}_3+2x_4+8x_5+x_2\bar{x}_3+6x_2\bar{x}_4+8x_3\bar{x}_5+3x_4\bar{x}_3+2x_5\bar{x}_4$$
$$=10+5x_3+5x_4+10x_5-x_2x_3-6x_2x_4-3x_3x_4-8x_3x_5-2x_4x_5.$$

We have

$$g_2 = -x_3 - 6x_4;$$

g_2 is negative if and only if at least one of x_3 and x_4 is equal to one, i.e.,

$$M_2 = \{(0, 1, \alpha_5', \alpha_6'), (1, 0, \alpha_5'', \alpha_6''), (1, 1, \alpha_5''', \alpha_6''')\}$$

(where α_5', α_6', α_5'', α_6'', α_5''', α_6''' may take arbitrary values in L_2), and is equal to zero if and only if $x_3 = x_4 = 0$, i.e.,

$$N_2 = \{(0, 0, \beta_5, \beta_6)\}$$

(where β_5, β_6 may take arbitrary values in L_2).

From formula (5) we see that

$$x_2 = \bar{x}_3 x_4 \cup x_3 \bar{x}_4 \cup x_3 x_4 \cup \bar{x}_3 \bar{x}_4 u_2,$$

or

$$x_2 = x_3 \cup x_4 \cup \bar{x}_3 \bar{x}_4 u_2; \tag{11.2}$$

taking $u_2 = 0$,

$$x_2^0 = x_3 \cup x_4,$$

and applying formula (1), we get

$$x_2^+ = x_3 + x_4 - x_3 x_4, \tag{11.2$^+$}$$

$$G_3 = 10 + 4x_3 - x_4 + 10x_5 - 3x_3x_4 - 6x_3x_5 - 2x_4x_5,$$

$$g_3 = 4 - 3x_4 - 6x_5;$$

g_3 is negative if and only if $x_5 = 1$, and is never equal to zero, hence,

$$x_3 = x_5,$$

$$G_4 = 10 - x_4 + 8x_5 - 5x_4x_5, \tag{11.3}$$

$$g_4 = -1 - 5x_5;$$

g_4 is negative for any x_5, hence applying (5′), we have,

$$x_4 = 1,$$

$$G_5 = 9 + 3x_5, \tag{11.4}$$

$$g_5 = 3;$$

g_5 is positive for any x_6; hence $M_5 = N_5 = \phi$, and

$$x_5 = 0; \tag{11.5}$$

analogously we have

$$x_6 = 0. \tag{11.6}$$

Hence

$$G_{\min} = G_6 = 9, \tag{12}$$

and solving (11.6), (11.5), (11.4), (11.3), (11.2), and (11.1) we get

$$x_1^0=1,\ x_2^0=1,\ x_3^0=0,\ x_4^0=1,\ x_5^0=0,\ x_6^0=0.$$

The minimal cut is

$$K=\{s_1,\ s_2,\ s_4\},\qquad K'=\{s_3,\ s_5,\ s_6\}, \tag{13}$$

and the corresponding maximal flows are:

f_{ij}	1	2	3	4	5	6
1	0	6	3	0	0	0
2	0	0	1	5	0	0
3	0	0	0	0	7	0
4	0	0	3	0	0	2
5	0	0	0	0	0	7
6	0	0	0	0	0	0

LOWER BOUNDS ON ARC FLOWS

LET US suppose that we are given a matrix $B=\|b_{ij}\|$ of real-valued elements called *lower bounds*, satisfying

$$0\leqq b_{ij}\leqq c_{ij}, \tag{14}$$

and we are seeking the maximal v for which f_{ij} satisfy (7) and

$$b_{ij}\leqq f_{ij}\leqq c_{ij}. \tag{15}$$

FORD AND FULKERSON[3] proved that:

If there is a matrix $\|f_{ij}\|$ satisfying (7) and (15) for some number v, then the maximal value of v subject to these constraints is equal to the minimum of

$$H=c(M,\ M')-b(M',\ M) \tag{16}$$

taken over all $M\subseteq N$ with $s_1\epsilon M$, $s_n\epsilon M'$.

It is easy to see that,

$$H=\textstyle\sum_{i=1}^{i=n}\sum_{j=1}^{j=n} c_{ij}\,x_i\,\bar{x}_j-\sum_{i=1}^{i=n}\sum_{j=1}^{j=n} b_{ij}\,\bar{x}_i\,x_j, \tag{17}$$

and as $x_1=1$, $x_n=0$ we have

THEOREM 2. *If there is a matrix* $\|f_{ij}\|$ *satisfying* (7) *and* (15) *for some* v, *then the maximal value of* v *subject to these constraints is equal to the minimum of the pseudo-Boolean function*

$$\begin{aligned}H=c_{1n}-b_{n1}+\textstyle\sum_{j=2}^{n-1}(c_{ij}\,\bar{x}_j-b_{nj}\,x_j)+\sum_{i=2}^{n-1}(c_{in}\,x_i-b_{i1}\,\bar{x}_i)&\\ +\textstyle\sum_{i=2}^{n-1}\sum_{j=2}^{n-1}(c_{ij}\,x_i\,\bar{x}_j-b_{ij}\,\bar{x}_i\,x_j).&\end{aligned} \tag{18}$$

Example. If in the previous example we take $b_{24}=3$, $b_{35}=4$, $b_{54}=2$, and all the other $b_{ij}=0$, and if we put

$$x_1=1, \tag{19.1}$$

then we have

$$H_2=10-2x_2+5x_3+2x_4+6x_5-x_2x_3-3x_2x_4+2x_2x_5-3x_3x_4-4x_3x_5-2x_4x_5;$$
$$g_2'=-2-x_3-3x_4+2x_5$$

and
$$x_2=\bar{x}_5\cup\bar{x}_3\bar{x}_4x_5u_2, \tag{19.2}$$
$$x_2^+=1-x_5, \tag{19.2$^+$}$$
$$H_3=8+4x_3-x_4+8x_5-3x_3x_4-3x_3x_5+x_4x_5;$$
$$g_3'=4-3x_4-3x_5,$$

and
$$x_3=x_4x_5. \tag{19.3}$$
$$H_4=8-x_4+8x_5-x_4x_5,$$
$$g_4'=-1-x_5,$$

and
$$x_4=1, \tag{19.4}$$
$$H_5=7+7x_5,$$
$$g_5'=7,$$

and
$$x_5=0. \tag{19.5}$$

We also have
$$x_6=0. \tag{19.6}$$

Solving the system (19.6), (19.5), (19.4), (19.3), (19.2), and (19.1) we obtain
$$x_1=1, \quad x_2=1, \quad x_3=0, \quad x_4=1, \quad x_5=0, \quad x_6=0, \tag{20}$$

and
$$H_{\min}=7. \tag{21}$$

A maximal flow in this case will be

f_{ij}	1	2	3	4	5	6
1	0	4	3	0	0	0
2	0	0	1	3	0	0
3	0	0	0	0	7	0
4	0	0	3	0	0	2
5	0	0	0	2	0	5
6	0	0	0	0	0	0

SOME FEASIBILITY THEOREMS

The Supply-Demand Theorem

Let $[N,A]$ be a network and let N be partitioned in sources S, intermediate nodes R, and sinks T. For any $s_i \epsilon S$ a nonnegative number α_i (the supply) is defined; for any $s_j \epsilon T$ a nonnegative number β_j (the demand) is defined.

Gale's theorem[6] may be now formulated as follows:

THEOREM 3. *The constraints*

$$\sum_{j=1}^{j=n} a_{ij} f_{ij} - \sum_{j=1}^{j=n} a_{ji} f_{ji} \begin{cases} \leqq \alpha_i & \text{if } s_i \epsilon S, \\ =0 & \text{if } s_i \epsilon R, \\ \geqq \beta_i & \text{if } s_i \epsilon T, \end{cases} \quad (22)$$

$$0 \leqq f_{ij} \leqq c_{ij}, \quad (23)$$

are feasible if and only if the minimum of the pseudo-Boolean function

$$\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} c_{ij}\, x_i\, \bar{x}_j + \sum_{h=1}^{h=n} \alpha_h\, \bar{x}_h - \sum_{h=1}^{h=n} \beta_h\, \bar{x}_h \quad (24)$$

is nonnegative.

A Symmetric Supply-Demand Theorem

Let us suppose that besides α_i and β_j we are also given some nonnegative numbers $\alpha_i'(s_i \epsilon S)$ and $\beta_j'(s_j \epsilon T)$ satisfying

$$0 \leqq \alpha_i \leqq \alpha_i', \qquad (s_i \epsilon S) \quad (25)$$

$$0 \leqq \beta_j \leqq \beta_j', \qquad (s_j \epsilon T) \quad (26)$$

and we are interested in the feasibility of the flow problem with the conditions:

$$\alpha_i \leqq \sum_{j=1}^{j=n} a_{ij} f_{ij} - \sum_{j=1}^{j=n} a_{ji} f_{ji} \leqq \alpha_i', \qquad (s_i \epsilon S) \quad (27)$$

$$\sum_{j=1}^{j=n} a_{ij} f_{ij} - \sum_{j=1}^{j=n} a_{ji} f_{ji} = 0, \qquad (s_i \epsilon R) \quad (28)$$

$$\beta_i \leqq \sum_{j=1}^{j=n} a_{ij} f_{ij} - \sum_{j=1}^{j=n} a_{ji} f_{ji} \leqq \beta_j', \qquad (s_i \epsilon T) \quad (29)$$

$$0 \leqq f_{ij} \leqq c_{ij}. \qquad (i, j = 1, \cdots, n) \quad (30)$$

Fulkerson's feasibility condition[4] now becomes:

THEOREM 4. *The relations* (27), (28), (29), *and* (30) *are feasible if and only if the minima of the pseudo-Boolean functions*

$$\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} c_{ij}\, x_i\, \bar{x}_j + \sum_{h=1}^{h=n} \alpha_h'\, \bar{x}_h - \sum_{h=1}^{h=n} \beta_h\, \bar{x}_h,$$

and

$$\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} c_{ij}\, x_i\, \bar{x}_j - \sum_{h=1}^{h=n} \alpha_h\, x_h + \sum_{h=1}^{h=n} \beta_h'\, x_h$$

are nonnegative.

Circulation Theorem

In networks with no source and sink specifications we are interested in the existence of *circulations* f_{ij} satisfying

$$\sum_{j=1}^{j=n} a_{ij} f_{ij} - \sum_{j=1}^{j=n} a_{ji} f_{ji} = 0, \quad (i = 1, \cdots, n) \quad (31)$$

$$b_{ij} \leqq f_{ij} \leqq c_{ij}. \qquad (i, j = 1, \cdots, n) \quad (32)$$

Hoffman's feasibility conditions[7] now becomes:

THEOREM 5. *The conditions* (31) *and* (32) *are feasible if and only if the minimum of the pseudo-Boolean function*

$$\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} c_{ij}\, x_i\, \bar{x}_j - \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} b_{ij}\, \bar{x}_i\, x_j \tag{33}$$

is nonnegative.

Another theorem of Hoffman[7] states

THEOREM 6. *The constraints* (32) *and*

$$\alpha_i \leqq \sum_{j=1}^{j=n} a_{ij} f_{ij} - \sum_{j=1}^{j=n} a_{ji} f_{ji} \leqq \alpha_i', \tag{34}$$

where $(0 \leqq \alpha_i \leqq \alpha_i',\ (i=1, \cdots, n)$ *are given real numbers) are feasible if and only if u being a free element of* L_2, *the minimum of the pseudo-Boolean function*

$$\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} c_{ij}\, x_i\, \bar{x}_j - \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} b_{ij}\, \bar{x}_i\, x_j - u \sum_{h=1}^{h=n} \alpha_h\, \bar{x}_h - \bar{u} \sum_{h=1}^{h=n} \alpha_h'\, x_h \tag{35}$$

is nonnegative.

ACKNOWLEDGMENT

I AM greatly indebted to PROF. GEORGE B. DANTZIG for his help and encouragement in the publication of this paper. I am also very indebted to one of the referees whose interesting comments and criticisms were of real value in the elaboration of the final manuscript of this article.

REFERENCES

1. L. R. FORD, JR. AND D. R. FULKERSON, "Maximal Flow Through a Network," *Canad. J. Math.* **8,** 399–404 (1956).
2. ——— AND ———, "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," *Canad. J. Math.* **9,** 210–218 (1957).
3. ——— AND ———, *Flows in Networks,* Princeton University Press, 1962.
4. D. R. FULKERSON, "A Network Flow Feasibility Theorem and Combinatorial Applications," *Canad. J. Math.* **11,** 440–451 (1959).
5. ———, "Flows in Networks," in *Recent Advances in Mathematical Programming,* pp. 319–332, McGraw-Hill, New York, 1963.
6. D. GALE, "A Theorem on Flows in Networks," *Pacific J. Math.* **7,** 1073–1082 (1957).
7. A. J. HOFFMAN, "Some Recent Applications of the Theory of Linear Inequalities to Extremal Combinatorial Analysis," *Proc. Symposia on Appl. Math.* **10** (1960).
8. P. L. IVĂNESCU, "Programmation Polynomiale en Nombres Entiers," *C. R. Acad. Sci. Paris* **258,** 424–427 (1964).
9. ———, "Integer Polynomial Programming," Communication at the *International Symposium on Mathematical Programming, London,* July 6–10, 1964 (Mimeographed).

10. ——— AND I. ROSENBERG, "Application of Pseudo-Boolean Programming to the Theory of Graphs," *Zeitschrift für Wahrscheinlichkeitstheorie* **3,** 163–176 (1964).
11. ———, ———, AND S. RUDEANU, "On the Minimization of Pseudo-Boolean Functions" (in Rumanian), *Studii şi Cercetări Matematice* **14,** 359–364 (1963).
12. ———, ———, AND ———, "An Application of Discrete Linear Programming to the Minimization of Boolean Functions" (in Russian), *Revue de Math. Pures et Appl.* **8,** 459–474 (1963).
13. ——— AND S. RUDEANU, "On Solving Transportation Problems with the Egerváry Method," I (in Rumanian), *Comunicările Academiei RPR* **11,** 773–778 (1961).
14. ——— AND ———, "On Solving Transportation Problems with the Egerváry Method," II (in Rumanian), *Studii şi Cercetări Matematice* **14,** 59–67 (1963).