DOCUMENT RESUME

ED 046 080 EA 003 201

AUTHOR	de Werra, D.
TITLE	Construction of School Timetables by Flow Methods.
INSTITUTION	Waterloo Univ. (Ontario). Dept. of Management Sciences.
REPORT NO	WP-14
PUB DATE	[70]
NOTE	24p.
EDRS PRICE	EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS	Computer Programs, Flexible Scheduling, *Mathematical Models, *Scheduling, *School Schedules, Time Blocks

ABSTRACT

In this paper, a heuristic algorithm for constructing school timetables is described. The algorithm is based on an exact method that applies to a family of particular timetable problems. The procedure has been used to construct timetables for Swiss schools having about 50 classes, 80 teachers, and 35 weekly periods. Less than five percent of class/teacher meetings could not be scheduled at some period in the week. (Author)

CONSTRUCTION OF SCHOOL TIMETABLES BY FLOW METHODS

by

D. de Werra

Working Paper No. ~~10~~ 14

DEPARTMENT OF MANAGEMENT SCIENCES
UNIVERSITY OF WATERLOO
WATERLOO, ONTARIO

CONSTRUCTION OF SCHOOL TIMETABLES BY FLOW METHODS

D. de Werra

Department of Management Sciences
University of Waterloo
Waterloo, Ontario

Statement of The Problem

We consider that a school is characterized by:

(a) a set of classes $C = \{c_i;\ i = 1, \ldots, q\}$

(b) a set of teachers $M = \{m_j;\ j = 1, \ldots, p\}$

(c) a set of subjects $B = \{b_l;\ l = 1, \ldots, s\}$

(d) a set of periods $\overline{N} = \{n_k;\ k = 1, \ldots, n\}$

(e) a set of meetings $\mathcal{R}$ which have to occur during the week.

Each meeting lasts one hour (or one period) and involves one or more classes, one or more teachers and one or more subjects.

A set of meetings involving the same elements (i.e. the same class(es), teacher(s) and subject(s)) is called a combination.

A timetable consists then of an allocation of every meeting in $\mathcal{R}$ to a period of the week; in other words it is a partition of $\mathcal{R}$ into n subsets $H_1, \ldots, H_n$ where H_k is the subset of all meetings occurring at period n_k (or more simply at period k). This allocation has to satisfy certain conditions specified later.

Problem I

Let us first consider a particular case of the timetable problem. Suppose that every meeting of $\mathcal{R}$ involves exactly one teacher m_j and one class c_i, and neglect the subjects.

We can associate with the set $\mathcal{R}$ an array A (pxq) with non-negative integer entries a_{ij} where a_{ij} is the number of meetings referring to class c_i and teacher m_j. Every combination is now characterized by c_i, m_j and a_{ij}; we shall denote a combination by (c_i, m_j, a_{ij}) or simply c_i-m_j. A is usually called the requirement matrix.

If the only restriction to be considered during the construction is that no teacher or class is required at two places at once, we can solve the problem easily. Let α_i be the total number of meetings involving class c_i and β_j the number of meetings involved teacher m_j. So if no teacher and no class is involved in more than n meetings, in other words if:

$$\begin{aligned} &\alpha_i \leq n \qquad && i = 1, \ldots, q \\ &\beta_j \leq n \qquad && j = 1, \ldots, p \end{aligned} \qquad (1)$$

then the problem has at least one solution and we can construct a timetable. We consider a bipartite multigraph G with vertices $c_1, \ldots, c_q; m_1, \ldots, m_p$. Every combination c_i-m_j is represented by a_{ij} parallel edges joining c_i and m_j. α_i (or β_j) is the degree of vertex c_i (or m_j), i.e. the number of edges incident with this vertex.

A subset of meetings which can occur simultaneously is represented by a matching H in G (it is by definition a set H of edges such that no two edges of H are adjacent). A timetable consists then of a partition of the edges of G into n matchings $H_1, \ldots, H_n$.

The determination of these matchings is a flow problem; more precisely it is a sequence of n-1 problems of flow with minimal cost[5].

Problem II

In a real school we find many other requirements and we have to take them into account during the preparation of a timetable.

We first introduce unavailability constraints in the previous basic model of problem I. Let us suppose that teachers and classes are not necessarily free during n periods, but some of them are unavailable for certain periods (they may be either absent or involved in preassigned meetings). These constraints are described by arrays E and D.

$$E = (e_{ik}) \qquad i = 1, \ldots, q; \quad k = 1, \ldots, n$$

$$D = (d_{jk}) \qquad j = 1, \ldots, p; \quad k = 1, \ldots, n$$

where e_{ik} (or d_{jk}) is 1 if c_i (or m_j) is unavailable at period k and is 0 otherwise.

For all i we define the subset M_i of M as the subset containing all teachers who have to meet class c_i and for all j, C_j is defined as the subset of all classes having to meet teacher m_j.

E is said to be consistent if, for all i and k, $d_{jk} = 1$ for all j such that $m_j \in M_i \Rightarrow e_{ik} = 1$; similarly D is consistent if for all j and k $e_{ik} = 1$ for all i such that $c_i \in C_j \Rightarrow d_{jk} = 1$. This means that a class c_i has to be considered as unavailable at period k if all teachers who are to meet c_i are unavailable at period k (and similarly for m_j). In the remainder of the paper we shall deal only with consistent arrays E and D.

As previously, a bipartite multigraph G can be associated with the timetable problem, but we have now an additional condition on H_k: H_k must not include any edge adjacent to vertex c_i (or m_j) if e_{ik} (or d_{jk}) is equal to 1.

Solubility Propositions

Since teachers and classes play equivalent roles in the problem, we shall simply call them elements. Necessary and sufficient conditions of solubility for problem II are not obtained as easily as for problem I.

Obviously no element must be involved in a number of meetings greater than the number of its free periods:

$$n - \sum_{k=1}^{n} e_{ik} \geq \alpha_i \qquad i = 1, \ldots, q$$
$$n - \sum_{k=1}^{n} d_{jk} \geq \beta_j \qquad j = 1, \ldots, p \tag{2}$$

Furthermore, it is necessary that for all combinations c_i-m_j the elements c_i and m_j are simultaneously free for a number of periods not less than a_{ij}:

$$a_{ij} \le \sum_{k=1}^{n} (1 - e_{ik})(1 - d_{jk}) \quad \text{for all } c_i - m_j \tag{3}$$

Simple examples show that conditions (2) and conditions (3) are not sufficient for solubility of problem II.

A third type of condition was used by Gotlieb(4); these are based on conditions of Hall for the existence of a system of distinct representatives(6). For all classes c_i and all subsets $\bar{M}$ of M_i, the number of meetings of c_i with the teachers of $\bar{M}$ must not be greater than the number of periods at which c_i and one at least of these teachers are simultaneously free:

$$\sum_{j:m_j \in \bar{M}} a_{ij} \le \sum_{k=1}^{n} (1-e_{ik})(1 - \prod_{j:m_j \in \bar{M}} d_{jk}) \tag{4.1}$$

for all $\bar{M} \subset M_i$ and for all i.

Similarly we have for all $\bar{C} \subset C_j$ and for all j:

$$\sum_{i:c_i \in \bar{C}} a_{ij} \le \sum_{k=1}^{n} (1-d_{jk})(1 - \prod_{i:c_i \in \bar{C}} e_{ik}) \tag{4.2}$$

These conditions imply conditions (2) and (3), but generally they are only necessary conditions. They are sufficient for a problem with only one teacher or one class.

We shall see in a following paragraph the meaning of conditions (4.1) and (4.2).

Let us call degree of freedom (DF) of c_i the quantity Mc_i defined as $Mc_i = n - \sum_k e_{ik} - \alpha_i$; and the DF of m_j will be denoted Mm_j and defined as

$$Mm_j = n - \sum_k d_{jk} - \beta_j$$

For combination c_i-m_j, we can also define a DF N_{ij} by

$$N_{ij} = \sum_k (1-e_{ik})(1-d_{jk}) - a_{ij}$$

With these definitions, conditions (2) and (3) express that all DF are nonnegative.

We give now some results concerning the solubility of problem II.

Proposition 1 A problem with $N_{ij} = 0$ for all combinations c_i-m_j may not have more than one solution.

Proof: If it had two distinct solutions there would be at least one combination, the a_{ij} meetings of which could be allocated in two different ways to a_{ij} periods of the week. This means that the number of periods where c_i and m_j are both free is greater than a_{ij}. This combination has a positive DF; this is in contradiction with $N_{ij} = 0$.

Proposition 2 If $N_{ij} = 0$ for all combinations c_i-m_j, then all DF of classes and teachers are nonpositive.

Proof: Let us associate with problem II a bipartite graph G^1. Its vertices are (fig. 1):

$$C_{ik} \quad i = 1, \ldots, q; \; k = 1, \ldots, n$$

$$M_{jk} \quad j = 1, \ldots, p; \; k = 1, \ldots, n$$

We call $\mathcal{C}_i$ the set $\{C_{ik}; k = 1, \ldots, n\}$ for $i = 1, \ldots, q$ and $\mathcal{M}_j$ the set $\{M_{jk}; k = 1, \ldots, n\}$ for $j = 1, \ldots, p$. C_{ik} and M_{jk} are joined by an edge if there is a combination c_i-m_j and if both c_i and m_j are free at period k.

(a) Since E and D are consistent, for any period k at which c_i is free, at least one teacher in M_i is free. This implies that there is at least one edge adjacent to C_{ik} if c_i is free, at period k. Similarly one edge at least is adjacent to M_{jk} if m_j is free at period k.

(b) Since $N_{ij} = 0$ for all c_i-m_j, there are exactly a_{ij} edges in G^1 joining vertices of $\mathcal{C}_i$ and vertices of $\mathcal{M}_j$ (there are exactly a_{ij} periods to which meetings of c_i-m_j can be allocated). So there are exactly α_i (or β_j) edges adjacent to vertices of $\mathcal{C}_i$ (or $\mathcal{M}_j$) in G^1.

(c) Suppose now that $Mc_i > 0$ for some class c_i: this means that the number (say α) of periods where c_i is free is strictly greater than the number α_i of meetings in which c_i is involved. Then (a) implies that there should be at least $\alpha > \alpha_i$ edges adjacent to vertices of $\mathcal{C}_i$ and (b) asserts that there are exactly α_i such edges;

this is a contradiction and thus all Mc_i are nonpositive. Clearly all Mm_j are also nonpositive and this ends the proof.

We know furthermore that it is always possible to associate with every timetable problem P_1 (having $Mc_i \geq 0$ and $Mm_j \geq 0$ for all i and j) another problem P_2 with $Mc_i = Mm_j = 0$ for all i and j. P_1 and P_2 are equivalent in the sense that one has a solution if and only if the other one is soluble(1)(5). P_2 is called a reduced problem and has the following properties: (1) the number of teachers is equal to the number of classes: (2) at any period k, the number of free classes is equal to the number of free teachers. So we only have to consider reduced problems.

<u>Proposition 3</u> A sufficient condition for solubility of a reduced problem II is $N_{ij} = 0$ for all c_i-m_j.

<u>Proof</u>: We consider the graph G^1: for all i, as $Mc_i = 0$, c_i is free during exactly α_i periods and by the consistency of E there are exactly α_i vertices of $\mathcal{C}_i$ adjacent to at least one edge: since there are exactly α_i edges adjacent to vertices of $\mathcal{C}_i$ ($N_{ij} = 0$ for all c_i-m_j) no vertex in $\mathcal{C}_i$ is adjacent to more than one edge. Similarly any vertex in $\mathcal{M}_j$ is adjacent to at most one edge, so that all edges of G^1 represent a timetable: at every period a class (or teacher) meets at most one teacher (or one class).

This result is worthy of consideration, because one normally thinks that the "probability" that a problem has a solution increases when the DF are augmented. This proposition shows that it is not always true: a reduced problem always has a solution if $N_{ij} = 0$, but it may have none when $N_{ij} \geq 0$.

Some special cases

Let us consider reduced problems; the unavailability constraints can be regarded as preassignments because at each period the number of unavailable classes is equal to the number of unavailable teachers (4)(5). So each element is to be involved in n meetings: some of these meetings are preassigned to fixed periods (corresponding to the unavailable periods of this element).

Proposition 5 A reduced problem such that one class only (or one teacher only) has some preassigned meetings, is soluble.

Proof: We consider the bipartite multigraph G associated with that problem; all edges representing preassigned meetings are adjacent. Thus any matching H in G contains at most one of these preassigned edges. So any decomposition of G into n matchings will give a timetable.

Note that if G is not connected, the problem is still soluble when at most one class (or one teacher) in each connected component has some preassigned meetings.

Now for a problem which is not reduced, let us define two families of unavailability constraints:

(a) all classes are free at any period, one teacher at most is not available at any period.

(b) one class exactly, say c_i, is unavailable at some periods. The unavailabilities of teachers are: at any period at which c_i is unavailable, one teacher at most is unavailable.

Similar constraints obtained by permuting classes and teachers are considered as equivalent to (a) or (b). Then we have the following result, provided that

$$\alpha_i + \sum_k e_{ik} \leq n \quad \text{for all } i \quad \text{and} \quad \beta_j + \sum_k d_{jk} \leq n \text{ for all } j$$

<u>Corollary 5.1</u> A problem II with unavailability constraints of type (a) or (b) has always a solution.

<u>Proof</u>: In case (a) we introduce a spurious class c_i^* ; if a teacher is unavailable at period k, we say he has to meet c_i^* at period k. In case (b) if c_i and m_j are both unavailable at period k, we say that c_i has to meet m_j at period k. All these preassigned meetings are represented by adjacent edges in G for (a) and for (b). G whose vertices have degrees not greater than n can be decomposed into n matchings; these matchings represent a timetable.

Of course the problem has still a solution when we have constraints (a) or (b) for any connected component of G.

Now let us examine the case where conditions of Hall hold for all classes and all teachers. This means that for a reduced problem the preassignments are such that it is possible to construct separate timetables for all classes and all teachers. The problem may have no solution; if this happens we can keep fixed all preassigned meetings of one class c_i (or of one teacher m_j) and allow the remaining preassigned meetings to be possibly allocated to some different periods. This is equivalent to consider that we have preassigned meetings only for one class c_i or for one teacher m_j. From proposition 5 such a problem always has a solution. Thus we can formulate:

Corollary 5.2 Let P be a reduced problem with a set of preassigned meetings. Suppose conditions of Hall hold for all classes and teachers. Then either P has a solution or it can be transformed into a soluble problem P' by keeping fixed the preassigned meetings of any class c_i (or any teacher m_j) and possibly modifying the remaining preassignments.

Flow Method

The method used to construct timetables proceeds by successive steps; the timetable will be prepared one period at a time; each step consists of the allocation of meetings to a determined period of the week. When we have completed the allocation for the k first periods we call the allocation of the remaining meetings to periods k+1, k+2, ..., n the residual problem. We have then to allocate meetings at period k in such a way that the residual problem still admits a solution. As any subset H_k of meetings that can occur simultaneously is represented by a matching in a bipartite multigraph for problem I as well as for problem II, we have to solve a flow problem at each step. To perform this it is convenient to consider the graph G associated with problem I; we orient any edge joining c_i and m_j from c_i to m_j and give it an infinite capacity. We introduce a source S and join it with every c_i by an edge (S, c_i) of unit capacity; similarly we introduce a sink T and edges (m_j, T) of unit capacities for all m_j. We call this graph G'. Any integer valued flow in G' from S to T corresponds to a matching in G.

At each step we have to assign priorities to meetings, and possibly to classes and teachers in order that the residual problem has a solution. These priorities can be expressed by a cost function KF associating a cost KF(x,y) with any edge (x,y) in G'. The allocation of meetings to a certain period is now a minimal cost flow problem.

For problem I the determination of a cost function KF is easy because we have necessary and sufficient conditions of solubility. At any step, we can for example assign costs equal to zero to all edges (S,c_i) and (m_j,T) associated with c_i and m_j which are still to be involved in a maximum number of meetings and positive costs to the remaining edges in G'. After each step we have to delete in G' the edges (c_i,m_j) corresponding to meetings which have just been allocated; the flow has to be set equal to zero in all remaining edges before proceeding to the next step.

As we do not know necessary and sufficient conditions for the solubility of problem II, we can examine different cost functions corresponding to different priority criteria.

I. Cost function F_1

For any period, we calculate the DF of classes and teachers:

$$KF(S,c_i) = \begin{cases} \infty \text{ if } c_i \text{ is unavailable at that period} \\ Mc_i \text{ otherwise} \end{cases}$$

$$KF(m_j,T) = \begin{cases} \infty \text{ if } m_j \text{ is unavailable at that period} \\ Mm_j \text{ otherwise} \end{cases}$$

$$KF(c_i,m_j) = \text{any positive number.}$$

Priorities are given to teachers and classes having the smallest DF. F_1 is a generalization of the cost function used in problem I. This cost function seems to be adapted to problems with only few unavailabilities.

II. Cost function F_2

At any period, we only calculate the DF of combinations.

$$KF(S,c_i) = \begin{cases} \infty \text{ if } c_i \text{ is unavailable at that period} \\ \text{any positive number otherwise} \end{cases}$$

$$KF(m_j,T) = \begin{cases} \infty \text{ if } m_j \text{ is unavailable at that period} \\ \text{any positive number otherwise} \end{cases}$$

$$KF(c_i,m_j) = N_{ij}$$

We try to allocate first meetings of combinations having the smallest DF. If there are no or only a few unavailabilities, we can find examples where F_2 fails to construct a solution while F_1 does. Consider the example in Fig. 2; such a problem obviously has a solution since no vertex is adjacent to more than 3 edges.

At the first step we have:

$$KF(I,a) = KF(I,b) = KF(I,c) = 2$$
$$KF(II,a) = KF(III,b) = KF(IV,c) = 1$$

With the flow method we get $H_1 = \{(II,a), (III,b), (IV,c)\}$.

The residual problem has no solution because class I has still to be involved in 3 meetings and there are only 2 periods left.

III. Cost function F_3

The mixed cost function F_3 is defined for any period as follows:

$$KF(S,c_i) = \begin{cases} \infty \text{ if } c_i \text{ is unavailable at that period} \\ Mc_i \text{ otherwise} \end{cases}$$

$$KF(m_j,T) = \begin{cases} \infty \text{ if } m_j \text{ is unavailable at that period} \\ Mm_j \text{ otherwise} \end{cases}$$

$$KF(c_i,m_j) = N_{ij}$$

With this cost function giving priority to elements and meetings having the smallest DF, we try to prevent these DF from decreasing too rapidly and if we can proceed up to the nth step with nonnegative DF we get a solution for problem II. If some DF becomes negative during the procedure, then the residual problem has no solution. This situation may happen even though problem II is soluble because the procedure is not based on necessary and sufficient conditions for solubility.

Some differences between problem I and problem II are to be noted now. In problem I, we could determine at each step a maximal flow (with minimal cost) corresponding to a maximal matching (i.e. containing a maximum number of edges). In other words we could allocate as many meetings as possible at any period; the residual problem remained soluble. But for problem II, it may happen that the residual problem becomes insoluble if we determine an allocation represented by a maximal flow (with minimal cost).

At a given period we have to allocate exactly one meeting of every combination c_i-m_j with $N_{ij} = 0$ (if c_i and m_j are both free); if we do not, the residual problem certainly has no solution. H_k has to include a subset of edges corresponding to combinations whose DF are zero and moreover a subset of edges adjacent to all vertices c_i and m_j (free at that period k) with DF equal to zero. An admissible H_k will thus sometimes be a non maximal matching; the consequence is that we cannot always get a timetable with a sequence of maximal flows. The example in Fig. 3 illustrates this fact. If we use F_3 and determine a maximal flow with minimal cost at first step, we cannot get a solution.

Practically we shall at any period allocate first meetings of all combinations c_i-m_j with $N_{ij} = 0$ (if c_i and m_j are free) and then determine a maximal flow with minimal cost corresponding to the meetings involving the remaining classes and teachers.

Application to a Real System

Many other requirements are present in real schoo.s, so that we still have to modify our model and introduce the related constraints:

(1) Let B* be a subset of B; the elements of B* are called special subjects. Meetings involving a subject b_e in B* have to take place in a determined room associated with this subject. As the schools are supposed to include exactly one room for any special subject, two meetings with the same special subject b_e may not occur simultaneously.

(2) As previously mentioned, a meeting may involve several teachers, several classes and several subjects.

(3) It is furthermore necessary that certain subjects are acceptably spaced throughout the week. So we want no more than one meeting of any combination to occur in a school day. In some particular cases, two meetings of the same combination must occur at consecutive periods. These are called double meetings; for every combination the number of double meetings is fixed.

We explicitly take these constraints into account as follows. Before constructing the timetable by the flow method, we allocate double meetings, meetings involving several teachers and classes, and meetings including special subjects at different periods without violating any constraints.

We eliminate then from Ω all meetings already assigned and consider that elements involved in these preassigned meetings are unavailable at the corresponding periods.

The remaining problem is now a problem II and we determine a solution by the flow method (usually with cost function F_3).

In practice, more than 5% of meetings are rarely preassigned, so that this rather arbitrary preassignment does not influence strongly the construction of a solution. The above outlined method is still a reliable procedure.

A program has been written in FORTRAN IV (with a few subroutines in MAP) for the IBM 7040 of the Centre de Calcul de l'Ecole Polytechnique Fédérale de Lausanne. We used Input and Output subroutines of a program written in Germany (H.G. Genrich[3]).

A series of experiments was carried out with the program. Because of the imposed constraints it was not always possible to avoid failure in constructing the timetables. Generally when n steps are performed, a small percentage of meetings are left unassigned. These are advantageously allocated with a minimum of manual adjustments. Some results are summarized in Table 1.

TABLE 1

Classes	Teachers	Periods of the Week	No. of Meetings	Percentage not Assigned	Computation Time
9	15	36	300	0.3	4 min.
34	64	35	1200	3	30 "
48	84	35	1700	5	50 "

REFERENCES

(1) J. Csima, Investigations on a Timetable Problem, Ph.D. Thesis, Institute of Computer Science, University of Toronto, 1965.

(2) L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton University Press, 1962.

(3) H.G. Genrich, Die Automatische Aufstellung von Stundenplänen auf relationentheoretische Grundlage, Forschungsbericht No. 1713, Landes Nordrhein-Westfalen, Köln, 1967.

(4) C.C. Gotlieb, The Construction of Class-Teacher Timetables, Proceedings of the IFIP Congress, North-Holland Publishing Co., Amsterdam, 1963, p. 73.

(5) D. de Werra, Résolution de problèmes d'horaire par la théorie des graphes, Thèse de doctorat, Institut de Mathématiques Appliquées, Ecole Polytechnique Fédérale de Lausanne, 1969.

(6) P. Hall, On representatives of subsets, J. London Math. Soc. 10, 1935, p. 26-30.

ACKNOWLEDGEMENTS

The author wishes to thank Professor Ch. Blanc (Swiss Federal Institute of Technology in Lausanne) who suggested this research and Professor D.J. Clough (University of Waterloo) for his helpful comments on an earlier version of this paper.

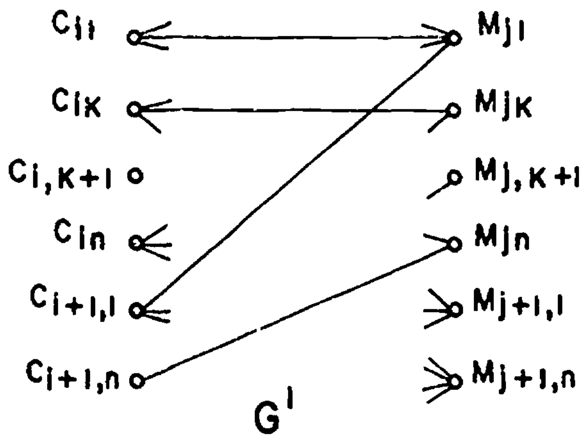

Fig. 1

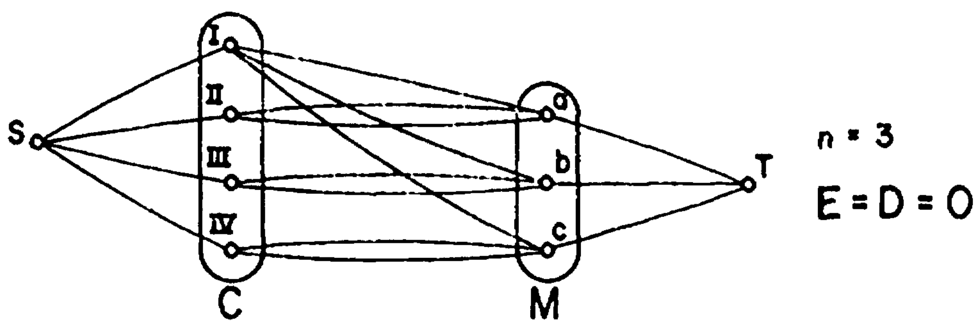

Fig. 2

$$A = \begin{array}{c} C_I \\ C_{II} \end{array} \begin{bmatrix} 1 & \\ 1 & 1 \end{bmatrix} \qquad n = 3 \qquad E = \begin{bmatrix} & & 1 \\ & 1 & \end{bmatrix} \qquad D = \begin{bmatrix} & & 1 \\ & 1 & \end{bmatrix}$$

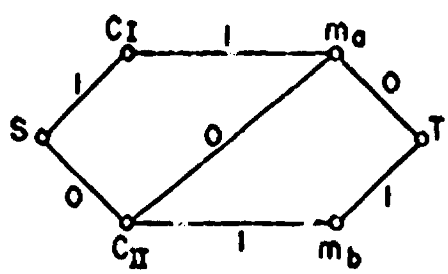

Fig. 3