INFORMATION SCIENCES 7, 95–132 (1974)

Networks of Constraints: Fundamental Properties and Applications to Picture Processing*

UGO MONTANARI

Istituto di Elaborazione della Informazione del C.N.R., Pisa, Italy

ABSTRACT

The problem of representation and handling of constraints is here considered, mainly for picture processing purposes. A systematic specification and utilization of the available constraints could significantly reduce the amount of search in picture recognition. On the other hand, formally stated constraints can be embedded in the syntactic productions of picture languages. Only binary constraints are treated here, but they are represented in full generality as binary relations. Constraints among more than two variables are then represented as networks of simultaneous binary relations. In general, more than one equivalent (i.e., representing the same constraint) network can be found: a minimal equivalent network is shown to exist, and its computation is shown to solve most practical problems about constraint handling. No exact solution for this central problem was found. Anyway, constraints are treated algebraically, and the solution of a system of linear equations in this algebra provides an approximation of the minimal network. This solution is then proved exact in special cases, e.g., for tree-like and series-parallel networks and for classes of relations for which a distributive property holds. This latter condition is satisfied in cases of practical interest.

1. INTRODUCTION

In writing this paper we had in mind mainly the problems of a particular field, namely picture recognition and description. However, the problem of proper representation and economic handling of constraints is very general and is important in many problems of operations research, engineering, and computer science. For instance, many practical design problems consist of finding any solution which satisfies all topological and geometrical restrictions [1]. Even when an optimization problem must be stated, the chosen constraint representation is essential in determining the nature of the mathematical prob-

*This work was carried out while the author was visiting at the Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pa. and was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107).

lem involved and its difficulty. Unfortunately, many practical constraints are difficult to handle, because they involve in a complicated way many variables. For instance, we can mention the noncrossing condition among electrical paths in an integrated circuit layout or, as a more esoteric example, the restriction to be faced in the design of computer rooms that all magnetic tape units must be in sight from the operator.

In picture processing, constraints play an important role, but they are unlikely to be representable in a linear or anyway simple form. Here constraints are better known by the name of (geometrical, topological, structural) *properties* of the class of pictures under consideration. But in fact they are present as fixed characteristics in explicit models or are implied by recognition routines which do not take into account configurations without the desired property.

However, we believe that an explicit and consistent treatment of constraints can bring valuable advantages. To show what we have in mind, we present some scenarios.

(A) SPEED UP OF THE RECOGNITION PROCESS

Often recognition subroutines search a picture for specific elements or features. The search space is usually more than two-dimensional, because other free parameters must be determined at the same time (e.g., the angular position of a stroke, the vertex structure in a cube [2]). For efficiency, what we are looking for must in general be dependent on what we have already found out about the particular picture. More precisely, it is useless to look for features which are possible *a priori*, but are not consistent with the part of the picture we have already recognized. For instance, if the problem is to recognize human faces [3], we must, of course, limit the search for particular elements (eyes, nose, mouth, ears, etc.) to the areas of the picture where they may ever be present. A second step is to establish constraints between *pairs* of elements. If for instance the position of one ear has already been determined, the area in which the mouth could be found is further restricted. Such binary constraints, if formally stated, can be intersected and composed. For instance, if also an eye has been determined, the allowed area for the mouth can be considered the intersection of the constraints given by the ear and the eye. Furthermore, the presence of elements yet unfound, such as the nose, but for which constraints have been defined, could transmit further constraints from determined elements to the sought ones. In conclusion, if all those constraints are superimposed, the search space can be reduced. Only the first few elements will be time consuming. For the others, the recognition procedure should be essentially a check of the evidence we have already gathered.

(B) OPTIMAL RECOGNITION

In particularly bad cases, combined evidence from all elements is required before accepting a picture, because many acceptable candidates are present for each feature. This situation arises mainly during preprocessing or anyway during recognition of unstructured entities (see [4] for the limit case of optimal detection of curves). In this case, it is convenient to assign a merit figure to the various alternatives, and then to find the best one with an optimization procedure. Again, systematic handling of constraints is vital in reducing the combinatorics involved.

(C) IMPERFECT MODELS

A model can be imprecise because it is too simple, or because something present in the model is missing in reality. In the former case, the model will not be as powerful as it could be, but it will work; while in the latter case the picture could be rejected as not satisfying the model. The missing part could be simply obscured by the noise. If the model is organized in terms of constraints, a model without the critical part could be systematically built, taking into account the constraints transmitted from one part of the model to the other through the missing part.

(D) LINGUISTIC METHODS

The application of parsing mechanisms in picture analysis is very promising [5, 6]. These methods work well if the structure of the image is mainly topological, as in bubble chamber tracks, chemical structures, block diagrams [7, 8]. If geometrical information is essential, it can be embedded in the syntax rules only in simple cases, as in the linguistic description of mathematical formulas [9]. If the allowed geometrical relations can be expressed as a set of simultaneous constraints, they can be formally added to the rewriting rules of the grammar under the form of applicability conditions [10]. In this way, perhaps the geometrical consistency of the various rules can be proved at a grammar level. This fact would guarantee that all the pictures *generated* by the grammar are consistent and representable on the plane.

In this paper, we have limited our formalization to binary constraints. On the other hand, they are represented in the most general way, i.e., as algebraic relations between sets of possible values of pair of variables [14]. For many variables, a constraint is then represented as a network of simultaneous binary relations.

Of course, an n-ary constraint cannot always be represented exactly by an n-vertex network of binary constraints. However, an optimal approximating network can be given easily. On the other hand, many different but equivalent networks can represent the same n-ary constraint. All the networks equivalent to a given one can be ordered by set inclusion. A least element is proved to exist and it is called the minimal network. Minimal networks are shown to have all the constraints as explicit as possible.

The problem of the determination of the minimal network from a given one is then shown to include most of the practical problems mentioned above about constraint composition and transmission. Unfortunately, no general algorithm was found. This is not surprising, since very tough problems, like the graph-coloring problems, fit this scheme.

Approximate solutions are considered instead. In these networks (which are called closed) all those global constraints are explicit that can be transmitted through all the possible paths in the network. The problem of determining a closed, equivalent network is then stated algebraically. If the operations of intersection and composition of constraints are defined, the above problem can be shown to be equivalent to the solution of a system of linear equations in this algebra. Since composition does not distribute, in general, over intersection, an iterated Gaussian elimination algorithm is required for finding the solution of the system.

In the last section, some special cases are investigated, in which closed networks are minimal, i.e., in which our algorithm computes the exact solution. If the topology of the network is restricted, tree-like and series-parallel closed networks are proved minimal. The same result is also achieved if we restrict the class of allowed relations to a class where composition distributes over intersection. This is the case when the sets of possible values of variables have a lattice structure and the relations satisfy a monotonicity assumption. This condition is satisfied in some cases of practical interest, such as the shortest path problem in a graph (possibly with negative weights) and its multimensional equivalents. Here, our algorithm becomes the well-known Floyd algorithm [12].

2. UNION, INTERSECTION, AND COMPOSITION OF CONSTRAINTS

In this section, a constraint between two variables is represented, in complete generality, by a relation between two sets. Elementary algebraic properties of relations are then recalled.

If a constraint exists between two variables x_1 and x_2, $x_1 \in X_1 = \{x_{1,1}, \ldots, x_{1,N_1}\}$, $x_2 \in X_2 = \{x_{2,1}, \ldots, x_{2,N_2}\}$ then in general not all possible pairs $(x_{1,r}, x_{2,s})$ are allowed. The set of allowed pairs is called a *relation* between sets X_1 and X_2. In general, it is convenient to consider ordered pairs and thus to distinguish between a relation R_{12} and a relation R_{21}. For instance, if $X_1 = \{1, 2\}$ and $X_2 = \{1, 2, 3\}$, then $R_{12} = \{(1, 1), (2, 1), (1, 3)\}$ is a relation. Any relation R_{12} is thus a subset, proper or improper, of the product set $X = X_1 \times X_2$ of all the pairs. A standard way of representing subsets is to use the characteristic function F

$$F\colon X_1 \times X_2 \longrightarrow \{0, 1\};\ F((x_{1,r}, x_{2,s})) = 1 \text{ iff } (x_{1,r}, x_{2,s}) \in R_{12}.$$

In other words, to each pair in X a binary digit is associated, which is 1 if and only if the pair belongs to the relation. Being characterized by a binary number of $N_1 \cdot N_2$ digits, $2^{N_1 \cdot N_2}$ different relations exist between X_1 and X_2. Usually, these digits are arranged in a $N_1 \times N_2$ matrix $[R_{12,rs}]$ whose rows correspond to the elements of set X_1 and columns to set X_2. Thus we have

$$R_{12,rs} = 1 \text{ iff } (x_{1,r}, x_{2,s}) \in R_{12}.$$

In our example, the characteristic binary matrix is:

$$R_{12} = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{vmatrix}.$$

In what follows, relations will be mainly represented in matrix notation.

The inverse R_{12}^{-1} of a relation R_{12} is defined as the transpose $R_{12,rs}^{-1} = R_{12,rs}^{T} = R_{21,sr}$.

For relations, being sets, we can define the usual operations of negation

$$\overline{R}_{12} = \neg R_{12} \text{ iff } \overline{R}_{12,rs} = \neg R_{12,rs} \quad (r = 1, \ldots, N_1; s = 1, \ldots, N_2)$$

union or logical sum

$$R_{12} = R'_{12} \cup R''_{12} \text{ iff } R_{12,rs} = R'_{12,rs} \vee R''_{12,rs}$$

intersection or logical product

$$R_{12} = R'_{12} \cap R''_{12} \text{ iff } R_{12,rs} = R'_{12,rs} \wedge R''_{12,rs}$$

and the partial ordering relation of set inclusion

$$R'_{12} \subseteq R''_{12} \text{ iff } R'_{12,rs} \subset R''_{12,rs}.$$

An empty relation ϕ_{12} and an universal relation U_{12} can be defined

$$\phi_{12,rs} = 0;\ U_{12,rs} = 1$$

such that

$$R_{12} \cup \phi_{12} = R_{12};\ R_{12} \cap U_{12} = R_{12},$$

for all R_{12}. Thus relations between two sets form a complete lattice with greatest element U and least element ϕ and where the operations of sup and inf coincide with union and intersection, respectively.

Next step is to consider a constraint R_{12} between variables x_1 and x_2 and a constraint R_{23} between variables x_2 and x_3. There will be an induced or transmitted constraint R_{13} between variables x_1 and x_3: a pair $(x_{1,r}, x_{3,s})$ is allowed, if at least one value $x_{2,t}$ exists, such that both $(x_{1,r}, x_{2,t})$ and $(x_{2,t}, x_{3,s})$ are allowed by R_{12} and R_{23}, respectively. This requirement defines the operation of *composition* of relations:

$$R_{13} = R_{12} \cdot R_{23} \text{ iff } R_{13,rs} = \bigvee_{t=1}^{N_2} R_{12,rt} \wedge R_{23,ts}.$$

Note that composition, in matrix notation, is just binary matrix multiplication. For example, we may have

$$R_{12} = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{vmatrix}; R_{23} = \begin{vmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{vmatrix}; R_{13} = R_{12} \cdot R_{23} = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}.$$

It is very easy to see that composition is associative, and that an identity relation exists, defined only between a set and itself,

$$I_{11,rs} = 1 \text{ iff } r = s$$

such that

$$R_{12} \cdot I_{22} = I_{11} \cdot R_{12} = R_{12}$$

for every relation R_{12}.

The defined operations of union, intersection and composition have an useful monotonicity property. If $f(R_{12})$ is any expression involving the operations of intersection, union and composition among relation R_{12} and any number of constants, from $R'_{12} \subseteq R''_{12}$ we have $f(R'_{12}) \subseteq f(R''_{12})$. This property is obvious if we notice that function f, written in binary form, contains binary sums and products, but no negations.

A particular case of a relation happens when one of the two sets (say the first) has just one element. These relations, in binary form, are representable as vectors and are in a one-to-one correspondence with the subsets of the second set. Actually, in what follows we will always assume the existence of a fictitious one-element set X_0, to have an homogeneous way of representing subsets. Especially useful in our formalism are the *fundamental vectors* V_{01} and V_{10}, i.e., the vectors with only one nonzero element. For instance, the *image* in R_{12} of the element $x_{1,r}$ can be defined as represented by the vector

$$R_{02} = V_{01} \cdot R_{12},$$

where V_{01} is the fundamental vector corresponding to element $x_{1,r}$

$$V_{01,t} = 1 \text{ iff } t = r.$$

A relation R_{12} is called *total* if every element of X_1 and X_2 is in relation with some other element. In our formalism, R_{12} is total iff

$$V_{01} \cdot R_{12} \neq \phi_{02} \text{ and } R_{12} \cdot V_{20} \neq \phi_{10}$$

for every fundamental vector V_{01} and V_{20}. Given any total relation R_{12}, it is easy to see that $R_{12} \cdot U_{23} = U_{13}$ and $U_{11} \cdot R_{12} = U_{12}$.

In what follows we are mainly interested in the operations of intersection and composition, so we will use the symbol + for intersection and the simple concatenation for composition. Unfortunately, composition does *not* distribute over intersection. In general

$$R_{12}\,(R'_{23} + R''_{23}) \neq R_{12}\,R'_{23} + R_{12}\,R''_{23}.$$

For instance, if

$$R_{12} = \begin{vmatrix} 1 & 1 \\ 0 & 0 \end{vmatrix};\; R'_{23} = \begin{vmatrix} 0 & 0 \\ 1 & 0 \end{vmatrix};\; R''_{23} = \begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix}$$

then

$$R_{12}\,(R'_{23} + R''_{23}) = \phi_{13};\; R_{12}\,R'_{23} + R_{12}\,R''_{23} = \begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix}.$$

A special case arises when distributivity does hold. In general, we say that the relations

$$R_{ik}, R_{ki} \quad i = 1, \ldots, n;\; i \neq k.$$

form a distributive set of relations with respect to set X_k if

$$\left(\sum_{\substack{i=1 \\ i \neq k}}^{m} V_{0i}\,R_{ik}\right)\left(\sum_{\substack{i=1 \\ i \neq k}}^{m} R_{ki}\,V_{i0}\right) = \sum_{\substack{i=1 \\ i \neq k}}^{m} \sum_{\substack{j=1 \\ j \neq k}}^{m} V_{0i}\,R_{ik}\,R_{kj}\,V_{j0} \tag{2.1}$$

for every set of fundamental vectors V_{0i}, V_{i0} ($i = 1, \ldots, m; i \neq k$) and for every m. In (2.1) the indexes of the sums go from 1 to m. Actually, the ordering is immaterial and thus we require (2.1) to hold whenever the indexes assume any set of m values. Note that distributivity defined in terms of fundamental vectors is more general than simple distributivity. For instance, from

$$V_{01}\,R_{12}\,(R'_{23}\,V'_{30} + R''_{23}\,V''_{30}) = V_{01}\,R_{12}\,R'_{23}\,V'_{30} + V_{01}\,R_{12}\,R''_{23}\,V''_{30}$$

for every V_{01}, V'_{30} and V''_{30}, (2.2) follows, but not conversely.

$$R_{12}\,(R'_{23} + R''_{23}) = R_{12}\,R'_{23} + R_{12}\,R''_{23}. \tag{2.2}$$

3. NETWORKS OF CONSTRAINTS

In this section, constraints among n ($n > 2$) variables are considered. A straightforward formalization of such constraints as n-ary relations is possible, but the quantity of information involved grows exponentially with n, and so no hope exists to handle it for any practical n. Networks of binary relations as defining an n-ary relation are then introduced. An optimal approximation theorem is proved, and just one minimal network is shown to exist. Finally, a problem is stated, called the central problem, which embodies most practical problems posed by networks of constraints.

Generalizing the approach followed in Section 2, it is clear that an n-ary constraint can be considered to allow some (or none or all) among the possible n-tuples of values of n variables. Thus an n-ary relation ρ is any subset of $X = X_1 \times X_2 \times \ldots \times X_n$. The set X can be visualized as an n-dimensional space. An n-ary relation ρ thus represents a "solid" in this space. Given an m-dimensional subspace $S = X_{i_1} \times \ldots \times X_{i_m}$, any n-tuple a and any n-ary relation ρ in X can be projected on S yielding the m-tuple a_S and the m-ary relation ρ_S. The number of distinct n-tuples is $N_1 \ldots N_n$ and thus $2^{N_1 \cdots N_n}$ is the number of distinct n-ary relations. If $N_1 = \ldots = N_n = N$ then N^n bits are required on the average for storing a n-ary relation. Practical values for N and n in picture-processing applications are 1000 and 20, and thus the information involved is enormous. One way out is to consider a restricted class of n-ary relations.

A network R of binary relations is defined as a set of sets $\overline{X} = \{X_1, \ldots, X_n\}$ plus a relation R_{ij} from every set X_i to every set X_j $(i, j = 1, \ldots, n)$. Furthermore, $R_{ii} \subseteq I_{ii} (i = 1, \ldots, n)$. If $R_{ij} = R_{ji}^T$, the network will be called *symmetric*. The network of relations R can be thought of as representing an n-ary relation

$$\rho = \{a \mid a \in X = X_1 \times \cdots \times X_n; (\forall i, j) S = X_i \times X_j; a_S \in R_{ij}\}.$$

In other words, an n-tuple a is allowed by ρ iff its projections on all the two-dimensional subspaces S of X simultaneously satisfy the binary constraints of the network R. Note that if some $R_{ij} = \phi_{ij}$ then $\rho = \phi$, while, if $\rho = X$ then $R_{ij} = U_{ij} (i, j = 1, \ldots, n; i \neq j)$ and $R_{ii} = I_{ii} (i = 1, \ldots, n)$.

An obvious way of visualizing a network is by a directed graph. Vertices $V_1, \ldots, V_n$ correspond to sets $X_1, \ldots, X_n$, and an arc $V_i V_j$ is present from V_i to V_j iff $R_{ij} \neq U_{ij} (i \neq j)$ or $R_{ii} \neq I_{ii}$. Relation R_{ij} is then associated with the direct arc $V_i V_j$. For instance the following n-ary relation ρ is represented by the network R in Fig. 1a:

$$\rho = \{(x_{1,1}, x_{2,1}, x_{3,1}), (x_{1,1}, x_{2,1}, x_{3,2}), (x_{1,2}, x_{2,3}, x_{3,1})\}$$

or, in a more compact notation,

$$\rho = \begin{vmatrix} 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \end{vmatrix} \tag{3.1}$$

where the indexes of the allowed n-tuples form the columns.

A network R is determined by giving in orderly fashion all its binary relations, and this requires roughly $B = n^2 N^2$ bits. Clearly $B < N^n$, except for very small values of N and n. This argument shows that the class of n-ary relations repre-

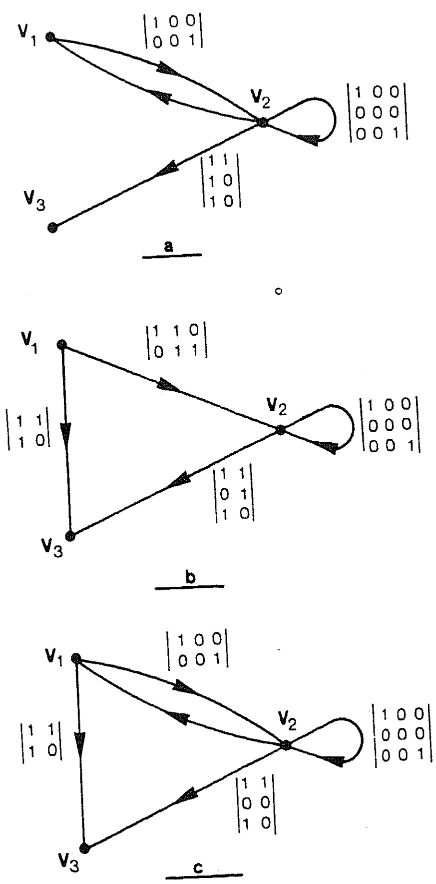

Fig. 1. Examples of networks of constraints. As a graphical convention, if both arcs V_iV_j and V_jV_i exist, but only V_iV_j is labeled with relation R_{ij}, then arc V_jV_i is assumed labeled with R_{ij}^T. Networks (a), (b), and (c) are equivalent, and network (c) is the intersection of networks (a) and (b).

sentable by networks is narrower (in fact, much narrower) than the class of all n-ary relations.

Given an n-ary relation ρ, the simple projection formula (3.2) generates a network R' which is, in a sense, the best possible excess approximation of ρ.

$$R'_{ij} = \{a_S \mid a \in \rho \text{ and } S = X_i \times X_j\}. \tag{3.2}$$

In words, if ρ is expressed in column form, R'_{ij} is obtained by taking the i-th and the j-th rows (and merging repeated pairs). Note that $R'_{ii} \subseteq I$ and that $R'_{ij} = R'^{T}_{ji}$, i.e., R' is symmetric.

Some properties of R' are proved by the following theorem.

THEOREM 3.1. *The network of relations R' defined by* (3.2) *represents an n-ary relation ρ' such that*

$$\rho \subseteq \rho'. \tag{3.3}$$

Furthermore, no network R'' exists, which represents an n-ary relation ρ'' such that

$$\rho \subseteq \rho'' \subset \rho'.$$

Thus, in particular, $\rho' = \rho$ if ρ is representable by a network.

Proof. Formula (3.3) is easy proved, because n-tuples $a \in \rho$ satisfy network R' by construction. We will prove the second part by contradiction. Let a be an n-tuple such that $a \in \rho'$ but $a \notin \rho''$. Thus, some projection b of a does not satisfy a relation of R'', say $b = a_S$, $S = X_i \times X_j$, $b \notin R''_{ij}$, while $b \in R'_{ij}$. But if the pair b was included in R'_{ij} it means that an n-tuple $\bar{a} \in \rho$ exists, such that $\bar{a}_S = b$. This is a contradiction, because then $\bar{a} \notin \rho''$, while we assumed $\rho \subseteq \rho''$.

Q.E.D.

As an example of the projection procedure let us consider the following relation:

$$\rho = \begin{vmatrix} 1 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 2 & 1 \end{vmatrix}. \tag{3.4}$$

The approximating network is:

$$R'_{12} = \begin{vmatrix} 1 & 1 & 2 \\ 1 & 2 & 2 \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix}; \; R'_{23} = \begin{vmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix};$$

$$R'_{13} = \begin{vmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}; \; R'_{11} = R'_{22} = R'_{33} = I.$$

The induced n-ary relation is

$$\rho' = \begin{vmatrix} 1 & 1 & 2 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 1 \end{vmatrix}. \tag{3.5}$$

A partial ordering among networks of constraints having the same number n of vertices can be introduced in a natural way. The ordering relation is defined as follows:

$$R' \subseteq R'' \text{ iff } R'_{ij} \subseteq R''_{ij} \; (i, j = 1, \ldots, n). \tag{3.6}$$

The reflexive, weakly antisymmetric and transitive properties for network inclusion descend from the same properties for set inclusion. It is also clear that the set of all networks with n vertices is a lattice under $\subseteq$ because a least ($R_{ij} = \phi_{ij}$) and greatest ($R_{ij} = U_{ij}$ if $i \neq j$; $R_{ii} = I$) networks exist. Thus union and intersection between two networks are defined. It is also immediate to see that

$$R' \subseteq R'' \text{ implies } \rho' \subseteq \rho'' \tag{3.7}$$

where R' and R'' represent ρ' and ρ''.

We have seen that not all the n-ary relations are representable by a network of constraints. It can also happen that an n-ary relation ρ is representable by many distinct networks. For instance, relation (3.1) can be represented by the networks in both Fig. 1a and Fig. 1b. Two networks R' and R'' which represent the same n-ary relation ρ are called *equivalent*.

The next theorem proves the existence of a *minimal* network M representing ρ.

THEOREM 3.2. *Let S_ρ be the equivalence class of all networks representing the same relation ρ. If*

$$R' \in S_\rho \textit{ and } R'' \in S_\rho$$

also

$$R = R' \cap R'' \in S_\rho.$$

(See, for instance, Fig. 1a, b, c). *As a consequence, a minimal (with respect to $\subseteq$) network M representing ρ exists, and can be obtained from ρ by the projection formula* (3.2).

Proof. To prove this theorem, we first notice that $R \subseteq R'$ and $R \subseteq R''$, and thus an n-tuple a satisfying R satisfies also R' and R'' for (3.7). Conversely, if a satisfies R' and R'' it satisfies also R. In fact, for each subspace $S = X_i \times X_j$, if $a_S \in R'_{ij}$ and $a_S \in R''_{ij}$ then $a_S \in R' \cap R'' = R$ by the definition of intersection. Finally, the network obtained by formula (3.2) must be minimal: if any pair b is erased by any relation R_{ij}, the represented relation ρ is changed. Q.E.D.

Given an n-ary relation ρ representable with a (minimal) network of constraints M, and a subspace $S = X_1 \times \ldots \times X_m$ of X, one could ask if the projection ρ_S of ρ:

$$\rho_S = \{a_S \mid a \in \rho\} \tag{3.8}$$

is representable with a network of m nodes. Interestingly enough, in the general case the answer is no, and a counterexample is given in Fig. 2. There, relation ρ is

$$\rho = \{(x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}), (x_{1,1}, x_{2,2}, x_{3,2}, x_{4,2}),$$
$$(x_{1,2}, x_{2,2}, x_{3,1}, x_{4,3})\}.$$

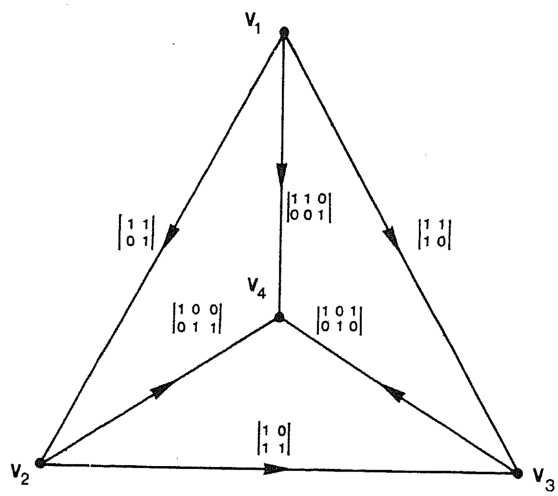

Fig. 2. An example of indecomposable network.

If $S = X_1 \times X_2 \times X_3$, ρ_S is given by (3.4). But, as we saw, (3.4) is not representable with a three-vertex network. If ρ_S is representable for all S, then ρ and all networks representing ρ are called *decomposable.* If not, the next theorem gives the best approximating network for ρ_S.

THEOREM 3.3. *The best, minimal approximating network of ρ_S is the complete subnetwork $M^{\bar{S}}$ of M corresponding to the set of vertices $\bar{S} = \{X_1, \ldots, X_m\}$.*

Proof. This property descends immediately from Theorem (3.1) and from the fact that if $T = X_i \times X_j$ is any bidimensional subspace of S, we have:

$$(\rho_S)_T = \rho_T. \qquad \text{Q.E.D.}$$

In the remainder of the paper we will be often concerned in proving that a network R is minimal: $M = R$. The next theorem gives a characteristic condition for R to be minimal.

THEOREM 3.4. *A necessary and sufficient condition for a network R to be minimal, is that if a pair b satisfies the generic relation R_{ij}, an n-tuple a satisfying R exists, such that $a_S = b$, $S = X_i \times X_j$.*

Proof. Necessity. If R is minimal and $b \in R_{ij}$, an n-tuple, $a \in \rho$, must exist such that $a_S = b$, because otherwise R' with $R'_{ij} = R_{ij} - \{b\}$ would be equivalent to R and smaller. *Sufficiency.* Under our assumption, no pair can be erased from any R_{ij} still obtaining an equivalent network. Thus R is minimal. Q.E.D.

The last theorem can be modified as follows.

COROLLARY. *Given any relation R_{ij} of R, if whenever a pair b belongs to R_{ij}, an n-tuple a satisfying R exists such that $a_S = b$, $S = X_i \times X_j$, then $R_{ij} = M_{ij}$.*

Proof. This statement follows from the previous theorem and from the equivalence of R and M. Q.E.D.

The above theorem shows that a minimal network of constraints is perfectly explicit: as far as the pair of variables x_i and x_j is concerned, the rest of the network does not add any further constraint to the direct constraint M_{ij}. Minimal networks are likely to represent an n-ary relation in a redundant way. In our application, for instance, we expect to define constraints almost only between geometrically adjacent elements. As a result, the density d of connections (defined as the average number of arcs per vertex) should be bounded, like that of a planar graph, $d < 3$[1] or at most should grow logarithmically with the number of vertices, but not linearly like in a complete graph.

From the above reasoning it should be clear that, in our application, networks of constraints will never be given or stored as minimal networks. Furthermore, the trivial way of getting the minimal network, i.e., generating the n-ary relation ρ from the given R, and then M from ρ with (3.2) will be impossible in continuous cases and always practically infeasible. Therefore, the problem of computing M from R in an economic way is nontrivial. On the other hand, we can show that most of the practical problems arising from the use of networks of constraints can be naturally reduced to the *central problem* of deriving M. In fact, in the first scenario described in the introduction, if V_{0k} $(k = 1, \ldots, m)$ are the fundamental vectors corresponding to the already determined values of the first m variables, the intersection of images

$$R_{0p} = \sum_{k=1}^{m} V_{0k} M_{kp}$$

clearly represents the set of allowed values for the p-th variable. In scenario (b), if we want to eliminate a variable (related to m others variables) using a sequential optimization method, we must optimize the objective function separately for all the feasible m-tuples of related variables. The set of those m-tuples is ρ_S, if S is the subspace of the related variables. By Theorem (3.3) $M^{\overline{S}}$ is the best approximating network of ρ_S. Finally, in scenario (c), the minimal subnetwork $M^{\overline{S}}$ corresponding to the parts which are not missing constitutes the best reduced model.

---

[1] In a planar graph, the number n of vertices is related to the number a of arcs by the relation $a \leqslant 3n - 6$. Equality is achieved if all the faces are triangular.

4. APPROXIMATE SOLUTION OF THE CENTRAL PROBLEM

In this section, we consider the problem of computing the minimal network equivalent to a given network. No exact general algorithm, besides complete enumeration, was found. However, an approximate solution is given, which generates an equivalent "closed" network.

In a generic network of constraints, a certain pair $(x_{i,r}, x_{j,s})$ can be allowed by the direct relation R_{ij} (or also by R_{ji}, R_{ii}, and R_{jj}), but can be actually forbidden because it is not possible to give to all the other variables any set of values allowed by all the constraints. To recognize such pairs and erase them, namely to make explicit the global constraint, is the essence of the central problem. The central problem, in its generality, is very difficult. Graph-coloring problems, for instance, are very neatly represented by networks of constraints: relations are all of the type U–I, i.e., all pairs are allowed except those of the same color. The number of allowed colors (i.e., the cardinality of sets X_i) and the topology of the graph characterize the particular problem. For instance, Fig. 3 shows the network of constraints representing the problem of coloring a tetrahedron with three colors: an impossible task. However, it is difficult to recognize it with a sequence of local examinations of the network, and without "higher-order" reasonings. Needless to say, no hope exists to extend such tricks

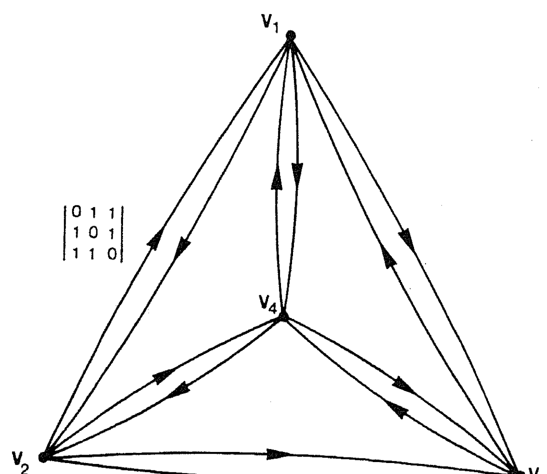

Fig. 3. In this network, the relation $\begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix}$ is associated to every arc. This network represents the impossible problem of coloring a four-vertex complete graph with three colors. This network is symmetric and closed but not minimal.

to the general case. Therefore, we look for an approximation of the minimal network M, i.e., a network Y which is as explicit as possible and still computable with local operations.

Let us consider an ordered pair of values

$$b = (x_{i,r}, x_{j,s})$$

and a path[2]

$$P = (V_i = V_{i_0}, \ldots, V_{i_p}, \ldots, V_{i_m} = V_j) \quad m \geqslant 1$$

in the complete network R from vertex V_i to vertex V_j. The pair b is *allowed by the path* P if the variables

$$x_i = x_{i_0}, \ldots, x_{i_p}, \ldots, x_{i_m} = x_j$$

can be given suitable values

$$x_{i,r} = x_{i_0, r_0}, \ldots, x_{i_p, r_p}, \ldots, x_{i_m, r_m} = x_{j,s}$$

which satisfy the relations

$$R_{i_0 i_1}, \ldots, R_{i_{p-1} i_p}, \ldots, R_{i_{m-1} i_m}$$

along the path P. Note that the same vertex V_k can occur in a path any number of times, and different values can be given to its variable x_k for each occurrence. A pair b is called *legal* if it is allowed by all the paths P from V_i to V_j. We will see that the property of being legal is decidable in a finite number of steps. Finally, a network is called *closed* if any pair b which is not legal is also not allowed by the direct relation R_{ij}.

It is clear from the definition that minimal networks are closed. The converse is, in general, not true. For instance, the network in Fig. 3 (representing the uncolorable tetrahedron) is closed but not minimal. This also means that many closed networks equivalent to a given network may exist. Given a network R, its *closure* Y is defined as the largest closed network not larger than R but equivalent to R. The next theorem proves the uniqueness of the closure.

THEOREM 4.1. *The set of closed networks not larger than R but equivalent to R (which is ordered under $\subseteq$) has a largest element Y. Therefore, Y is the only closure of R.*

Proof. We must prove that the union of two closed networks Y' and Y'', both not larger than R but equivalent to R, is a closed network Y not larger than R but equivalent to R. In fact from $R \supseteq Y'$ and $R \supseteq Y''$ we have $R \supseteq Y' \cup Y'' = Y$. From $R \supseteq Y \supseteq Y'$, R equivalent to Y' and (3.7) twice, we have Y equivalent

[2] A path in R is any sequence of vertices. A vertex can occur more than once in a path, even in consecutive positions.

to R. Let $Y_{ij,rs} = 1$. Then for $Y = Y' \cup Y''$ either $Y'_{ij,rs} = 1$ or $Y''_{ij,rs} = 1$ or both, say $Y'_{ij,rs} = 1$. Then the pair $b = (x_{i,r}, x_{j,s})$ is allowed by P in Y' for closure. Thus b is allowed by P also in Y, because the same set of path values satisfying Y' satisfies also Y, for $Y' \subseteq Y$.

Q.E.D.

The closure Y of a network R can be characterized as being the solution of the following system of equations.

$$Y_{ij} = \sum_{k=1}^{n} R_{ik} Y_{kj} + d_{ij} \tag{4.1}$$

where

$$d_{ij} = I_{ij} \text{ if } i = j;\ d_{ij} = U_{ij} \text{ otherwise.}$$

A network of relations Y is called *a solution* of system (4.1) iff:

(i) The relations Y_{ij} satisfy equations (4.1).

(ii) No other network Y' exists, such that Y' satisfies equations (4.1) and $Y' \supseteq Y$.

Note that condition (ii) is necessary for ruling out solutions which are not equivalent to R (like the trivial case $Y_{ij} = \phi_{ij}$) and that it does not imply uniqueness of the solution *a priori*. We can prove the following theorem.

THEOREM 4.2. *Any network Y which satisfies system* (4.1) *is:*

(a) Not larger than R.

(b) Closed.

If furthermore Y is a solution of system (4.1), *then Y is:*

(c) Equivalent to R.

(d) The closure of R.

Therefore only one solution exists.

Proof. (a) From (4.1) we have

$$Y_{jj} \subseteq I_{jj}.$$

Thus, by the monotonicity property of composition

$$R_{ij} Y_{jj} \subseteq R_{ij}.$$

And finally, from (4.1)

$$Y_{ij} \subseteq R_{ij} Y_{jj} \subseteq R_{ij}.$$

(b) Given any path P, we will prove that if $Y_{ij,rs} = 1$ and Y satisfies equations (4.1) then the pair $b = (x_{i,r}, x_{j,s})$ is allowed by P in R, i.e., the relations of R along P can be satisfied. We will prove this result by induction on the length m of the path P. If $m = 1$ the proof is trivial, because Y_{ij} is the only relation which must be satisfied. If the result is true for every path of length $(m - 1)$, it is true also for paths of length m. In fact, from (4.1) we have:

$$Y_{ij} = Y_{i_0 i_m} \subseteq R_{i_0 i_1} Y_{i_1 i_m}.$$

Therefore at least one value x_{i_1, r_1} must exist, such that $R_{i_0 i_1, r_0 r_1} = 1$ and $Y_{i_1 i_m, r_1 r_m} = 1$. Thus relation $R_{i_0 i_1}$ is satisfied by $(x_{i_0, r_0}, x_{i_1, r_1})$, while $Y_{i_1 i_m, r_1 r_m} = 1$ implies that the $m - 1$ relations $R_{i_1 i_2}, \ldots, R_{i_{m-1} i_m}$ can be satisfied according to the induction assumption.

(c) If an n-tuple satisfies Y, it satisfies also R, because $Y \subseteq R$ for (a). Conversely, let $a = (x_{1, r_1}, \ldots, x_{n, r_n})$ be an n-tuple satisfying network R, namely $R_{ij, r_i r_j} = 1$ $(i, j = 1, \ldots, n)$. We will prove that a satisfies network Y too. In fact, if we assume $Y_{ij, r_i r_j} = 0$ for some i, j, it is possible to find a larger network $Y' \supseteq Y$ (against (ii)) satisfying (4.1) and such that $Y_{ij, r_i r_j} = 1$. For proving it, let

$$Y^0_{ij, r_i r_j} = 1;\; Y^0_{ij, rs} = Y_{ij, rs} \text{ if } r \neq r_i \text{ and } s \neq r_j \; (i, j = 1, \ldots, n)$$

It is immediate to see that equations

$$Y^0_{ij, r_i r_j} = \bigwedge_{k=1}^{n} \left(\bigvee_{t=1}^{Nk} R_{ik, r_i t} \wedge Y^0_{kj, t r_j} \right) \wedge d_{ij, r_i r_j}$$

are satisfied, while from Y satisfying (4.1) we have:

$$Y^0_{ij, rs} \subset \bigwedge_{k=1}^{n} \left(\bigvee_{t=1}^{Nk} R_{ik, rt} \wedge Y^0_{kj, ts} \right) \wedge d_{ij, rs} \; (i, j = 1, \ldots, n, r \neq r_i, s \neq s_j)$$

from monotonicity of union and intersection. So we have

$$Y^0_{ij} \subseteq \sum_{k=1}^{n} R_{ik} Y^0_{kj} + d_{ij}.$$

Now, if we compute iteratively

$$Y^p_{ij} = \sum_{k=1}^{n} R_{ik} Y^{p-1}_{kj} + d_{ij}$$

we will have

$$Y^0 \subseteq Y^1 \subseteq \ldots \subseteq Y^p$$

from monotonicity of intersection and composition. Thus for some q we will have

$$Y^q = Y^{q+1} = Y'$$

satisfying (4.1) and such that $Y' \supseteq Y^0 \supset Y$.

(d) Let Y be the closure of R. From the definition of closed network, applied to paths of length two, we have

$$\overline{Y}_{ij} \subseteq \overline{Y}_{ik}\, \overline{Y}_{kj} + d_{ij}.$$

Then, from $Y \subseteq R$

$$\overline{Y}_{ij} \subseteq R_{ik}\, \overline{Y}_{kj} + d_{ij}.$$

Summing up, we have

$$\overline{Y}_{ij} \subseteq \sum_{k=1}^{n} R_{ik}\, \overline{Y}_{kj} + d_{ij}.$$

Actually, we must have equality, i.e. (4.1), because otherwise with the iterative method shown in part (c), a network $\overline{Y}'$ can be found which satisfies (4.1). Then the proof of part (c) shows that a network $\overline{Y}'' \supseteq \overline{Y}' \supseteq \overline{Y}$ can be found which satisfies (4.1) (and thus is closed and not larger than R) and is equivalent to R. The existence of $\overline{Y}''$ would contradict the maximality of $\overline{Y}$ proved in Theorem (4.1). Thus $\overline{Y}$ satisfies (i). On the other hand, if Y is any solution, we must have $\overline{Y} \supseteq Y$ for the same reason. Thus $\overline{Y}$ satisfies (ii) and is the only solution. Q.E.D.

The next corollary gives a simple way for characterizing a closed network.

COROLLARY. *A necessary and sufficient condition for a network Y to be closed is to satisfy the following system of equations*

$$Y_{ij} = \sum_{k=1}^{n} Y_{ik}\, Y_{kj} + d_{ij} \quad (i, j = 1, \ldots, n). \tag{4.2}$$

Proof. According to Theorem (4.2b), if Y satisfies system (4.2) Y is closed. Conversely, if Y is closed, it must be the closure of itself and thus by Theorem (4.2d) must satisfy equations (4.2). Q.E.D.

The next theorem proves a useful property of closed networks.

THEOREM 4.3. *In a closed network Y, the loop relations and the relations among different vertices satisfy the following equation*

$$Y_{ii} = Y_{ij}\, U_{ji} + I_{ii} \quad (i, j = 1, \ldots, n). \tag{4.3}$$

Proof. From (4.2) we have

$$Y_{ii} \subseteq Y_{ij}\, Y_{ji} + I_{ii}$$

and from monotonicity

$$Y_{ii} \subseteq Y_{ij}\, U_{ji} + I_{ii}.$$

Furthermore, from (4.2) we have

$$Y_{ij} \subseteq Y_{ii}\, Y_{ij}.$$

In binary form we have

$$Y_{ij,rs} \subset Y_{ii,rr} \wedge Y_{ij,rs} \quad (r = 1, \ldots, N_i; s = 1, \ldots, N_j)$$

or, from a truth table

$$Y_{ij,rs} \subset Y_{ii,rr}.$$

Making the union with respect to s

$$\bigvee_{s=1}^{N_j} Y_{ij,rs} \subset Y_{ii,rr} \quad (r = 1, \ldots, N_i).$$

Equivalently, we can write

$$Y_{ij}\, U_{ji} + I_{ii} \subseteq Y_{ii}.$$

Therefore we have

$$Y_{ii} = Y_{ij}\, U_{ji} + I_{ii}. \qquad \text{Q.E.D.}$$

Given a network R with n vertices we can give an algorithm for computing its closure Y.

Algorithm C
Step 1 $Y^0 = R$.
Step 2 Execute next step for $k = 1, \ldots, n$.
Step 3 $Y^k_{ij} = Y^{k-1}_{ij} + Y^{k-1}_{ik}\, Y^{k-1}_{kk}\, Y^{k-1}_{kj} \quad (i, j = 1, \ldots, n).$ (4.4)
Step 4 If $Y^n \neq Y^0$ then let $Y^0 = Y^n$ and go to Step 2; else let $Y = Y^n$ and stop.

THEOREM 4.4. *Algorithm C computes the closure Y of R. In particular, if $Y^n_{ij,rs} = 1$ in the network Y^n obtained at the end of the first iteration, then pair $(x_{i,r}, x_{j,s})$ is allowed by all the paths from V_i to V_j in R.*

Proof. We will prove that (i) Y is not larger than and equivalent to R; (ii) Y is closed; (iii) for every closed network Y' equivalent to R and not larger than R we have $Y' \subseteq Y \subseteq R$.

(i) Each application of Step 3 produces a network Y^k equivalent to the precedent Y^{k-1}. In fact, clearly $Y^k \subseteq Y^{k-1}$. On the other hand, if the second term in the right member of (4.4)

$$\bigvee_{t=1}^{N_k} Y^{k-1}_{ik,rt} \wedge Y^{k-1}_{kk,tt} \wedge Y^{k-1}_{kj,ts} = 0$$

is zero for some r, s it means that no value for x_k can be found which satisfies relations Y^{k-1}_{ik}, Y^{k-1}_{kk} and Y^{k-1}_{kj} for $x_i = x_{i,r}$ and $x_j = x_{j,s}$. Thus no n-tuple is excluded by Y^k which is not excluded by Y^{k-1}. Therefore, from transitivity of equivalence and inclusion, Y is equivalent to R and $Y \subseteq R$.

(ii) We will prove that if a pair is allowed by Y^n then is allowed by all paths in Y^0. We assume that when Step 3 was executed $(k - 1)$ times, if a pair $(x_{i,r}, x_{j,s})$ is allowed by the relation Y_{ij}^{k-1}, then it is allowed also by all the paths in Y^0 with extrema in V_i and V_j and having all the intermediate vertices with indexes $\leqslant k - 1$. If $k = 1$, the assumption is trivially true. We will prove the same property for k after the kth execution. Let us consider any path P from V_i to V_j having intermediate vertices with indexes $\leqslant k$. If vertex V_k does not belong to this path, the induction step is proved. If it does, path P can be decomposed in three paths:

(a) a path from V_i to V_k;
(b) a finite (possibly zero) number of circuits from V_k to V_k;
(c) a path from V_k to V_j.

All those paths have intermediate vertices with indexes $\leqslant k - 1$. According to the step (4.4), if $Y_{ij,rs}^k = 1$, then a value $x_{k,t}$ can be found such that $Y_{ik,rt}^{k-1} = 1$, $Y_{kk,tt}^{k-1} = 1$ and $Y_{kj,ts}^{k-1} = 1$. Thus, by the induction assumption, we can give to all the intermediate variables of paths (a), (b), and (c) suitable values which satisfy the corresponding constraints in Y^0. When the algorithm stops, we have $Y^n = Y^0 = Y$, and thus Y is closed.

(iii) Let $Y' \subseteq R$ be any closed network equivalent to R. We will have $Y' \subseteq Y^k$ for all k and for all iterations of algorithm C. Thus also $Y' \subseteq Y$. Inductively, let us assume that $Y' \subseteq Y^{k-1}$ before the execution of Step 3. This is certainly true for the first execution of Step 3 in the first iteration $Y^0 = R$. Then $Y' \subseteq Y^k$. In fact, if network Y' is closed, it satisfies equations (4.2) for the corollary to Theorem (4.2). In particular, we have

$$Y'_{ij} \subseteq Y'_{ik}\, Y'_{kj} + d_{ij};\; Y'_{kj} \subseteq Y'_{kk}\, Y'_{kj}$$

and therefore for the monotonicity of composition and intersection

$$Y'_{ij} \subseteq Y'_{ik}\, Y'_{kk}\, Y'_{kj} + d_{ij}$$

and thus

$$Y'_{ij} = Y'_{ij} + Y'_{ik}\, Y'_{kk}\, Y'_{kj} + d_{ij}\,. \tag{4.5}$$

But from the induction assumption and monotonicity we have

$$Y'_{ij} + Y'_{ik}\, Y'_{kk}\, Y'_{kj} + d_{ij} \subseteq Y_{ij}^{k-1} + Y_{ik}^{k-1}\, Y_{kk}^{k-1}\, Y_{kj}^{k-1} + d_{ij}. \tag{4.6}$$

Now note that the d_{ij} term in the right member is redundant because if $i \neq j$ then $d_{ij} = U_{ij}$ and if $i = j$ then $d_{ii} = I_{ii}$ and $Y_{ii}^{k-1} \subseteq R_{ii} \subseteq I_{ii}$. Thus from (4.4), (4.5), and (4.6) we have

$$Y'_{ij} \subseteq Y_{ij}^k \qquad \text{Q.E.D.}$$

It is interesting to see how the number of iterations required by algorithm C is dependent on the order of vertices in Step 2. For instance, in network R in

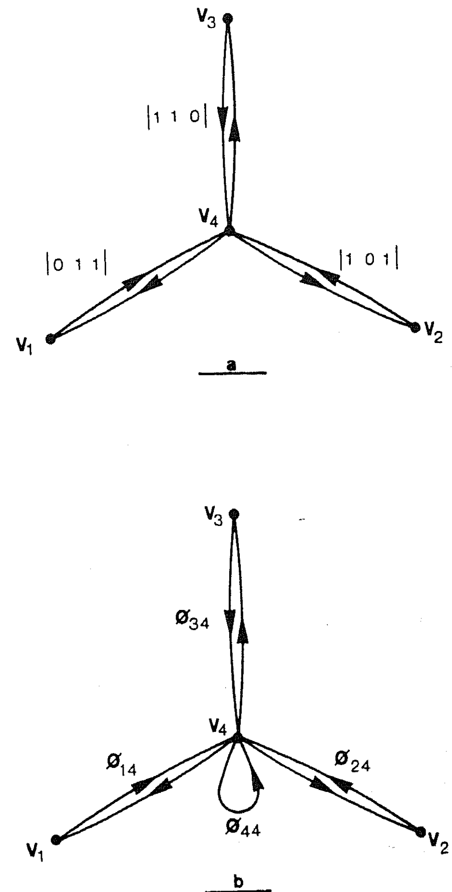

Fig. 4. (a) A symmetric network of constraints. (b) The network of constraints Y' equivalent to (a) computed by algorithm C in one iteration, with order of elimination (4, 1, 2, 3).

Fig. 4a, if $k = 1, 2, 3, 4$ then only one iteration is necessary for finding the closure $Y(Y_{ij} = \phi_{ij})$. If $k = 4, 1, 2, 3$ then two iterations are necessary. Fig. 4b shows the network Y' obtained at the end of the first iteration.

According to the above theorem, if $Y^n_{ij,\,rs} = 1$ in the network X, Y' obtained at the end of first iteration, then the pair $(x_{i,\,r}, x_{j,\,s})$ is allowed by all paths in R. For instance, in the above example for $i = 1, j = 2$, and $r = s = 1$ and for path $P = (V_1, V_3, V_4, V_2)$, values $x_{1,\,1}, x_{3,\,1}, x_{4,\,1}$ and $x_{2,\,1}$ satisfy the three relations R_{13}, R_{34}, R_{42} along P. Note how this condition is *not* sufficient for Y' being closed. Thus, in general, one iteration of algorithm C is not sufficient. On the other hand, each iteration of algorithm C produces an equivalent, strictly smaller network, and thus convergence is assured.

5. EXACT SOLUTION OF THE CENTRAL PROBLEM FOR PARTICULAR CLASSES OF NETWORKS

In the last section we have been able to give only an approximate solution to the central problem in the general case. A closed network instead of the minimal network was obtained. We can now ask if there are particular cases in which a closed network is always minimal.

In general, given a network R and a pair of vertices V_i, V_j we are interested in knowing if the relation Y_{ij} of the closure Y of R coincides with the relation M_{ij} of the minimal network equivalent to R. In this case, network R will be called *regular with respect to pair* $V_i V_j$. If R is regular with respect to all pairs of vertices, it will be called *regular*. Thus for a regular network R we have $Y = M$. In this section, we will see that interesting classes of networks are regular.

We can determine regular classes of networks in essentially two ways: either constraining the topology of the network or restricting the type of allowed relations. We will consider the former case first.

A symmetric[3] series-parallel network (spn) with respect to a pair of vertices $V_i V_j$ is usually defined recursively as follows

a) A complete symmetric network with two vertices V_i and V_j is a spn.
b) Given two spn's with respect to $V_i'V_j'$ and $V_i'' V_j''$, the network obtained letting V_j' and V_i'' coalesce is a spn with respect to $V_i' V_j''$.
c) Given two spn's with respect to $V_i' V_j'$ and $V_i'' V_j''$, the network obtained letting $V_i = V_i' = V_i''$ and $V_j = V_j' = V_j''$ is a spn with respect to $V_i V_j$.

As usual, all missing arcs $V_i V_j$ are assumed to correspond to the universal relations U_{ij}.

In the last section we saw that in a closed network Y a relation Y_{ij} makes explicit the constraints given by all the *paths* from V_i to V_j. The next lemma proves that a closed network has the same property for all the subnetworks which are series parallel with respect to $V_i V_j$.

LEMMA 5.1. *Let Y be a symmetric closed network, and let R be any subnetwork of Y which is a spn with respect to $V_i V_j$. Let M be the minimal network equivalent to R. We have*

$$Y_{ij} \subseteq M_{ij}. \tag{5.1}$$

---

[3]In what follows, symmetry will be almost always required, since a minimal network is obviously symmetric, while symmetry is not assured for a closed network. On the other hand, an equivalent, symmetric and not larger network R' can be immediately computed from any R with the formula: $R'_{ij} = R_{ij} + R^T_{ji}$. Its closure Y is then symmetric, as is obvious from algorithm C.

Proof. According to the corollary to Theorem (3.4) we must prove that if $Y_{ij,rs} = 1$ then an m-tuple satisfying R (R has m vertices) such that $x_i = x_{i,r}$ and $x_j = x_{j,s}$ can be found. We will use induction applied to each step of the recursive definition of a spn: we assume that the property is true for the component networks and we prove it for the resulting network. For steps of type (a), if $Y_{ij,rs} = 1$ then $Y_{ji,sr} = 1$ for symmetry, while $Y_{ii,rr} = 1$ and $Y_{jj,ss} = 1$ for (4.3). For steps of type (b), let V_k be the vertex in the middle of the series. If $Y_{ij,rs} = 1$ and Y is closed, then a value $x_k = x_{k,t}$ exists such that $Y_{ik,rt} = 1$ and $Y_{kj,ts} = 1$ and therefore the recursive assumption can be applied because no constraint exists in R between two vertices in different components. In definition (c), the recursive assumption can be directly applied to both components. Q.E.D.

As an example of application of this theorem, let us consider again the coloring problem represented by the closed but not minimal network Y in Fig. 3. In Fig. 5a, we have a subnetwork R which is a spn with respect to all pairs of vertices except $V_2 V_3$. In Fig. 4b, we have the minimal network M equivalent to R. Note that $Y_{ij} \subseteq M_{ij}$ (in fact, $Y_{ij} = M_{ij}$) for all pairs except $V_2 V_3$.

The next theorem proves the regularity of some classes of networks.

THEOREM 5.1. *(a) Tree networks*[4] *are regular. (b) Symmetrical series-parallel networks with respect to a pair $V_i V_j$, possibly with trees rooted at any vertex, are regular with respect to $V_i V_j$.*

Proof. (a) Let R be a tree network, and let Y be its closure. Given a pair $V_i V_j$, let P be the only connecting path in R. If $Y_{ij,rs} = 1$, suitable values can be given to the vertices of P which satisfy the relations of Y along P, since Y is closed. It is now easy to see that suitable values can also be given to the other variables. It is sufficient to evaluate them following the tree structure of R, i.e., in such a way that each new vertex V_k to be evaluated is adjacent in R to one already evaluated vertex V_h (more than one vertex is not possible: a circuit would be present). Inductively, let $x_{h,t}$ be the value already assigned to x_h. To x_k we can assign any value $x_{k,s}$ such that $Y_{hk,ts} = 1$. Such a value must exist, because otherwise $Y_{hh,tt} = 0$ for Theorem (4.3) and thus $Y_{ph,tt} = 0$ for all p (again for the same Theorem) so that the value $x_{h,t}$ could not have been given previously to x_h according to this procedure. At the end, all the variables have been evaluated and satisfy the Y constraints along the tree, and thus also all the R constraints, because $Y \subseteq R$ for Theorem (4.2(a)), and R has constants only along the tree. Therefore also Y is satisfied, because Y and R are equivalent. Thus Y is minimal and R is regular.

(b) Let R be the spn and let Y be its closure. Let Y' be the subnetwork of Y topologically equivalent to R. We have $Y \subseteq Y' \subseteq R$. But Y is equivalent to R

---

[4] As with spn, the branches of the tree correspond to symmetric, 2-vertex networks.

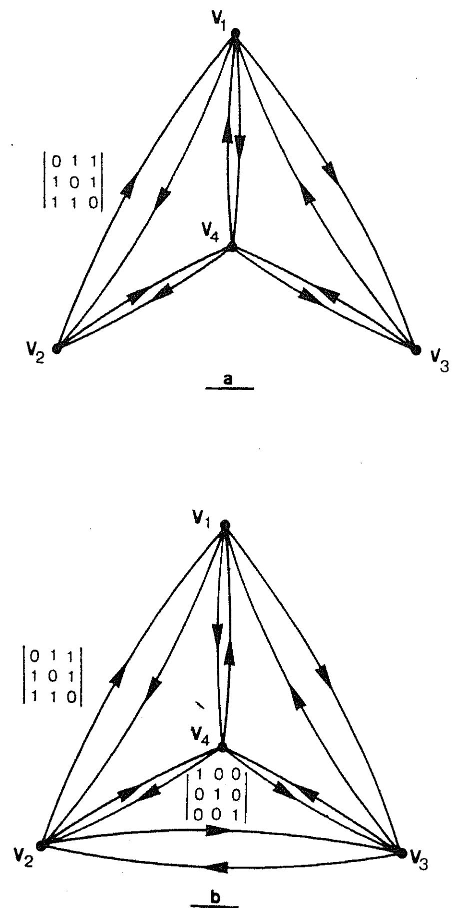

Fig. 5. (a) A series-parallel network with respect to all pairs of vertices except $V_2 V_3$. (b) Its minimal equivalent network. All nonlabelled arcs are assumed labeled with $\begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix}$.

and thus also Y' is equivalent to both Y and R from (3.7) twice. Now let M be the minimal network equivalent to Y' and thus to Y and R. For Lemma (5.1) we have $Y_{ij} \subseteq M_{ij}$. But M and Y are equivalent and M is minimal. Thus $M_{ij} = Y_{ij}$ and R is regular with respect to $V_i V_j$. If trees are present at some vertices of the spn, we can find a feasible n-tuple evaluating the vertices of the spn first, and then evaluating the vertices of each tree as in part (a) of this theorem.

Q.E.D.

In the remainder of this section, we want to determine a class of regular networks by restricting the type of allowed relations. We need the following definition. A network R such that relations R_{ik} and R_{ki} $(i = 1, \ldots, n; i \neq k)$ form a distributive set of relations (see condition (2.1)) for all k, is called a *distributive network*. We can now prove the following theorem.

THEOREM 5.2. *A closed, distributive network Y is decomposable. Furthermore, its symmetrization*

$$Y'_{ij} = Y_{ij} + Y^T_{ji} \; (i, j = 1, \ldots, n)$$

is minimal. Thus in particular if Y is symmetric, it is minimal.

Proof. We will prove first that if variables $x_1, \ldots, x_{k-1}$ can be given values $x_{1,r_1}, \ldots, x_{k-1,r_{k-1}}$ which satisfy relations Y_{ij} $(i, j = 1, \ldots, k-1)$, a value x_{k,r_k} can be given to variable x_k which, together with the previous values, satisfy relations Y_{ij} $(i, j = 1, \ldots, k)$. From the assumption we have $Y_{ij,r_ir_j} = 1$ or equivalently

$$V_{0i} Y_{ij} V_{j0} = U_{00} \quad i, j = 1, \ldots, k-1,$$

where the fundamental vectors V_{0i} and V_{j0} are defined as

$$V_{0i,r} = 1 \text{ and } V_{i0,r} = 1 \text{ iff } r = r_i$$

From the corollary to Theorem (4.2) we have

$$Y_{ij} \subseteq Y_{ik} Y_{kj} \quad i, j = 1, \ldots, k-1.$$

From monotonicity, we obtain:

$$V_{0i} Y_{ik} Y_{kj} V_{j0} = U_{00}.$$

Summing with respect to i and j

$$\sum_{i=1}^{k-1} \sum_{j=1}^{k-1} V_{0i} Y_{ik} Y_{kj} V_{j0} = U_{00}.$$

Applying distributivity (2.1) we have

$$\left(\sum_{i=1}^{k-1} V_{0i} Y_{ik}\right)\left(\sum_{j=1}^{k-1} Y_{kj} V_{j0}\right) = U_{00}.$$

Then a value x_{k,r_k} can be found, such that

$$\left(\sum_{i=1}^{k-1} V_{0i} Y_{ik}\right)_{,r_k} = 1 \text{ and } \left(\sum_{j=1}^{k-1} Y_{kj} V_{j0}\right)_{,r_k} = 1$$

or, completely in binary form,

$$Y_{ik,\, r_i r_k} = 1 \text{ and } Y_{kj,\, r_k r_j} = 1 \quad i,j = 1, \ldots, k-1.$$

Therefore, from Theorem (4.3) we have

$$Y_{kk,\, r_k r_k} = 1.$$

Finally, adding the inductive assumption, we have

$$Y_{ij,\, r_i r_j} = 1 \quad i,j = 1, \ldots, k.$$

Observing that the ordering of variables is immaterial and using the above proof as induction step, we have shown that if a k-tuple b satisfies any complete subnetwork $Y^{\overline{S}}$ of Y, $\overline{S} = \{V_{i_1}, \ldots, V_{i_k}\}$, at least one n-tuple a exists, whose projection on S is b, which satisfies Y. On the other hand, if b does not satisfy $Y^{\overline{S}}$, a does not satisfy Y by definition. Thus the projection ρ^S of the n-ary relation ρ represented by Y is representable by $Y^{\overline{S}}$ and therefore Y is decomposable. Furthermore, Y' is minimal. In fact, if $Y'_{ij,\, rs} = 1$ the ordered pair $b = (x_{i,\, r}, x_{j,\, s})$ satisfies the two-vertex subnetwork $Y^{\overline{S}}$, $\overline{S} = \{X_i, X_j\}$. Then by the first part of this theorem an n-tuple a exists, such that a satisfies Y and $a_S = b$. But Y is equivalent to Y' and thus a satisfies Y' as well. Therefore Y' is minimal for Theorem (3.4). Q.E.D.

In what follows, it is convenient to consider a particular case in which a slightly different distributive property holds. Given a network R, let us consider the set D of all the relations equal to all the possible expressions obtained by combining relations R_{ij} with the operations of intersection and composition. If in D right and left distributivity of composition over intersection always holds, R is called *star-distributive*. In this case, given any expression, it can always be reduced to a sum of products using distributivity. It is immediate to see that each term of the sum is the constraint represented by a path between the same pair of nodes. Therefore every relation D_{ij} in D represents the global constraint transmitted by some set of paths between vertices V_i and V_j. Especially interesting then are the limit relations D^*_{ij}[5] representing the global constraint transmitted by *all* the paths in R between V_i and V_j. D^* is the corresponding network, called *limit network.*

The next theorem proves some interesting properties of D^*.

THEOREM 5.3. *Let R be a star-distributive network, let D^* be its limit network, let Y^n be the network obtained after one iteration of algorithm C, and let Y be the closure of R. We have*

(a) $D^* = Y^n$.

(b) $D^* = Y$.

Therefore $Y = Y^n$ and one iteration is sufficient for algorithm C.

---

[5] As usual, we assume $D^*_{ii} \subseteq I_{ii}$ $(i = 1, \ldots, n)$.

Proof. (a) According to Theorem (4.4), if $Y^n_{ij,\,rs} = 1$, then the pair $(x_{i,\,r}, x_{j,\,s})$ is allowed by all the paths in R. Therefore

$$Y^n_{ij} \subseteq D^*_{ij}.$$

On the other hand, Y^n_{ij} is obtained, in algorithm C, with a finite number of intersections and compositions of relations of R. Therefore Y^n_{ij} is the sum of some of the terms of which D^*_{ij} is the sum: thus

$$D^*_{ij} \subseteq Y^n_{ij}.$$

Therefore we have

$$D^*_{ij} = Y^n_{ij} \; (i, j = 1, \dots, n).$$

(b) It is very easy to see that D^* is the solution of system (4.1). We prove first that D^* satisfies equations (4.1). In fact, let us consider the relation D^*_{ij}. It is equal to the sum of the terms corresponding to all paths between V_i and V_j. The first factor of each term must be one of the relations R_{ik} $(k = 1, \dots, n)$. Partitioning the paths and factorizing R_{ik}, we clearly obtain the right member

$$\sum_{k=1}^{n} R_{ik} D^*_{kj} + d_{ij}.$$

In fact, if $i = j$ the condition $D^*_{ii} \subseteq I_{ii}$ holds by construction. Since D^* satisfies (4.1), from Theorem (4.2) we have

$$D^* \subseteq Y.$$

But we have

$$Y \subseteq Y^n,$$

and, from part (a)

$$D^* = Y^n.$$

Thus we have also

$$D^* = Y. \qquad \text{Q.E.D.}$$

It may be interesting to see how in the star-distributive case algorithm C is nothing else that the solution by Gaussian elimination of the system of equations (4.1). We will show it with an example. If $n = 3$, we have

$$\begin{aligned} Y_{1j} &= R_{11} Y_{1j} + R_{12} Y_{2j} + R_{13} Y_{3j} + d_{1j} \\ Y_{2j} &= R_{21} Y_{1j} + R_{22} Y_{2j} + R_{23} Y_{3j} + d_{2j}. \quad j = 1, \dots, n \\ Y_{3j} &= R_{31} Y_{1j} + R_{32} Y_{2j} + R_{33} Y_{3j} + d_{3j}. \end{aligned}$$

Now it is easy to see from a truth table that the solution of the single equation

$$Z = AZ + B$$

is

$$Z = AB$$

if we are interested (as we are) only in the largest Z. Thus applying distributivity, the first equation of our system becomes

$$Y_{1j} = R_{11}\, d_{1j} + R_{11}\, R_{12}\, Y_{2j} + R_{11}\, R_{13}\, Y_{3j}.$$

Then substituting in the other two equations, multiplying and factorizing, we get

$$Y_{1j} = R_{11}\, d_{1j} + R_{11}\, R_{12}\, Y_{2j} + R_{11}\, R_{13}\, Y_{3j}$$

$$Y_{2j} = R_{21}\, R_{11}\, d_{1j} + (R_{22} + R_{21}\, R_{11}\, R_{12})\, Y_{2j} + (R_{23} + R_{21}\, R_{11}\, R_{13})\, Y_{3j} + d_{2j}$$

$$Y_{3j} = R_{31}\, R_{11}\, d_{1j} + (R_{32} + R_{31}\, R_{11}\, R_{12})\, Y_{2j} + (R_{33} + R_{31}\, R_{11}\, R_{13})\, Y_{3j} + d_{3j}.$$

The matrix of coefficients of this new system is exactly equal to Y^1 if we notice that $R_{11} \subseteq I_{11}$ and thus $R_{11} = R_{11}\, R_{11}$ and furthermore $R_{k1}\, R_{11} = R_{k1} + R_{k1}\, R_{11}$. After elimination of Y_{2j} and Y_{3j} we obtain

$$Y_{1j} = Y^3_{11}\, d_{1j} + Y^3_{12}\, d_{2j} + Y^3_{13}\, d_{3j}$$

$$Y_{2j} = Y^3_{21}\, d_{1j} + Y^3_{22}\, d_{2j} + Y^3_{23}\, d_{3j}$$

$$Y_{3j} = Y^3_{31}\, d_{1j} + Y^3_{32}\, d_{3j} + Y^3_{33}\, d_{3j}.$$

For example, if we write the first equation for $j = 2$, we have

$$Y_{12} = Y^3_{11}\, U_{12} + Y^3_{12} + Y^3_{13}\, U_{32}.$$

But it could be possible to see[6] that

$$Y^3_{ij} \subseteq Y^3_{ik}\, U_{kj}.$$

In conclusion we have

$$Y_{ij} = Y^3_{ij}.$$

We can also point out that algorithm C is similar to the Warshall algorithm [11] for finding the closure of a relation or to the Floyd algorithm [12] for determining the shortest path between all pairs of nodes in a weighted graph, or to the algorithm for deriving a regular expression from a left linear grammar or from a transition graph [13]. The similarity is not casual. In fact, all these algorithm can be considered the solution by Gaussian elimination of a linear sys-

[6] Actually, the terms of the type $Y^3_{ik}\, U_{kj}$ would not even exist, if variable elimination had taken place separately for the different values of index j.

tem of equations in a suitable algebra. We can find the same analogy in Jordan algorithm for matrix inversion in the usual linear algebra. The only difference is that in this case the solution of the single equation

$$Z = AZ + B$$

is

$$Z = (1 - A)^{-1} B,$$

while in our algebra, as already pointed out, the solution is

$$Z = AB.$$

In what follows, we impose restrictions on our relations for obtaining distributive and star-distributive networks.

Let us define a partial ordering $\leqslant$ in the set X_i of values of the variable x_i.[7] Furthermore, let us superimpose on X_i a complete lattice structure together with the operations of sup and inf. A total relation R_{ij} between a set X_i and a set X_j will be called *monotone* if it has the following properties:

(i) if $R_{ij,\,rs} = 1$ and $t \geqslant r$ then $R_{ij,\,ts} = 1$ and conversely
if $R_{ij,\,rs} = 1$ and $t \leqslant s$ then $R_{ij,\,rt} = 1$ (5.2)

(ii) if $R_{ij,\,ps} = 1, R_{ij,\,qs} = 1$ and $r = \inf(p, q)$ then $R_{ij,\,rs} = 1$ and
if $R_{ij,rp} = 1, R_{ij,\,rq} = 1$ and $s = \sup(p, q)$ then $R_{ij,\,rs} = 1$. (5.3)

The next theorem will clarify the kind of relations allowed by the above definition.

THEOREM 5.4. *Given a total relation R_{ij}, a necessary and sufficient condition for R to be monotone is that a defining function*

$$f_{ij} : X_i \longrightarrow X_j$$

exists, such that

$$R_{ij,\,rs} = 1 \text{ iff } s \leqslant f_{ij}(r) \tag{5.4}$$

and

$$f_{ij}(\inf(r_1, r_2)) = \inf(f_{ij}(r_1), f_{ij}(r_2)) \tag{5.5}$$

or, by duality, that an inverse defining function

$$g_{ij} : X_j \longrightarrow X_i$$

exists, such that

$$R_{ij,\,rs} = 1 \text{ iff } r \geqslant g_{ij}(s) \tag{5.6}$$

[7]For notational simplicity we will consider the partial ordering as defined on the set of indexes as well. For instance $r \leqslant s$ is equivalent to $x_{i,r} \leqslant x_{i,s}$.

and

$$g_{ij}(\sup(s_1, s_2)) = \sup(g_{ij}(s_1), g_{ij}(s_2)). \tag{5.7}$$

Proof. We will prove this theorem only for conditions (5.4) and (5.5). The proof in term of the inverse defining function is exactly dual. *Sufficiency.* From equation (5.5) we have

$$\text{if } r_1 \leqslant r_2 \text{ then } f_{ij}(r_1) \leqslant f_{ij}(r_2). \tag{5.8}$$

Therefore if $R_{ij,\,rs} = 1$ and $t \geqslant r$ from (5.4) and (5.8) we have

$$s \leqslant f_{ij}(r) \leqslant f_{ij}(t)$$

and thus

$$R_{ij,\,ts} = 1.$$

If $R_{ij,\,rs} = 1$ and $t \leqslant s$ we have

$$t \leqslant s \leqslant f_{ij}(r)$$

and thus

$$R_{ij,\,rt} = 1.$$

If $R_{ij,\,rp} = 1, R_{ij,\,rq} = 1$ and $s = \sup(p, q)$ we have

$$p \leqslant f_{ij}(r) \text{ and } q \leqslant f_{ij}(r)$$

and thus

$$s = \sup(p, q) \leqslant f_{ij}(r)$$

therefore

$$R_{ij,\,rs} = 1.$$

If $R_{ij,\,ps} = 1, R_{ij,\,qs} = 1$ and $r = \inf(p, q)$ we have

$$s \leqslant f_{ij}(p) \text{ and } s \leqslant f_{ij}(q)$$

and thus

$$s \leqslant \inf(f_{ij}(p), f_{ij}(q)) = f_{ij}(r)$$

therefore

$$R_{ij,\,rs} = 1.$$

Necessity. If a relation R_{ij} is monotone it can be put in the form (5.4). In fact, given an element r of X_i let us compute the superior $f_{ij}(r)$ of the image of r in R_{ij}. For (5.3) we have

$$R_{ij,\,rf_{ij}(r)} = 1.$$

Thus, for (5.2), equation (5.4) is satisfied. Function $f_{ij}(r)$ satisfies equation (5.5). In fact, for (5.2) we have

$$R_{ij,r_1 f_{ij}(\inf(r_1,r_2))} = 1 \text{ and } R_{ij,r_2 f_{ij}(\inf(r_1,r_2))} = 1$$

and thus, for the definition of f_{ij}

$$f_{ij}(\inf(r_1,r_2)) \leqslant f_{ij}(r_1) \text{ and } f_{ij}(\inf(r_1,r_2)) \leqslant f_{ij}(r_2)$$

Therefore

$$f_{ij}(\inf(r_1,r_2)) \leqslant \inf(f_{ij}(r_1), f_{ij}(r_2)). \tag{5.9}$$

On the other hand, we have

$$R_{ij,r_1 f_{ij}(r_1)} = 1 \text{ and } R_{ij,r_2 f_{ij}(r_2)} = 1.$$

From (5.2)

$$R_{ij,r_1 \inf(f_{ij}(r_1), f_{ij}(r_2))} = 1 \text{ and } R_{ij,r_2 \inf(f_{ij}(r_1), f_{ij}(r_2))} = 1.$$

Thus from (5.3)

$$R_{ij,\inf(r_1,r_2)\inf(f_{ij}(r_1), f_{ij}(r_2))} = 1.$$

Therefore

$$f_{ij}(\inf(r_1,r_2)) \geqslant \inf(f_{ij}(r_1), f_{ij}(r_2)).$$

Finally from the above relation and (5.9), equation (5.5) follows. Q.E.D.

A few examples will clarify the kind of relations allowed by the monotonicity constraint. For instance, if the partial ordering is also total, equation (5.5) can be substituted by equation (5.8), i.e., the defining function must be monotone. In Fig. 6a we see a monotone relation represented by a bipartite graph. We have $f_{ij}(5) = 4, f_{ij}(4) = f_{ij}(3) = 2, f_{ij}(2) = f_{ij}(1) = 1$. Conversely, $g_{ij}(1) = 1, g_{ij}(2) = 3, g_{ij}(3) = g_{ij}(4) = 5$. A special case of monotone relation, with infinite sets, is represented by the "shortest path" constraint

$$s \leqslant f_{ij}(r) = r + d.$$

In fact, the shortest path problem in a weighted graph is a special case of our central problem. The network of relations R can be obtained from the weighted graph as follows. The set of values for each variable is the set of natural numbers and all relations R_{ij} $(i, j = 1, \ldots, n)$ are monotone. If R_{ij} is specified by the defining function f_{ij}^R we have

$$x_j \leqslant f_{ij}^R(x_i) = x_i + t_{ij}$$

where t_{ij} are the arcs weights: $t_{ij} = t_{ji}, t_{ii} = 0$. We will see that the minimal network M has the same form

$$x_j \leqslant f_{ij}^M(x_i) = x_i + d_{ij}$$

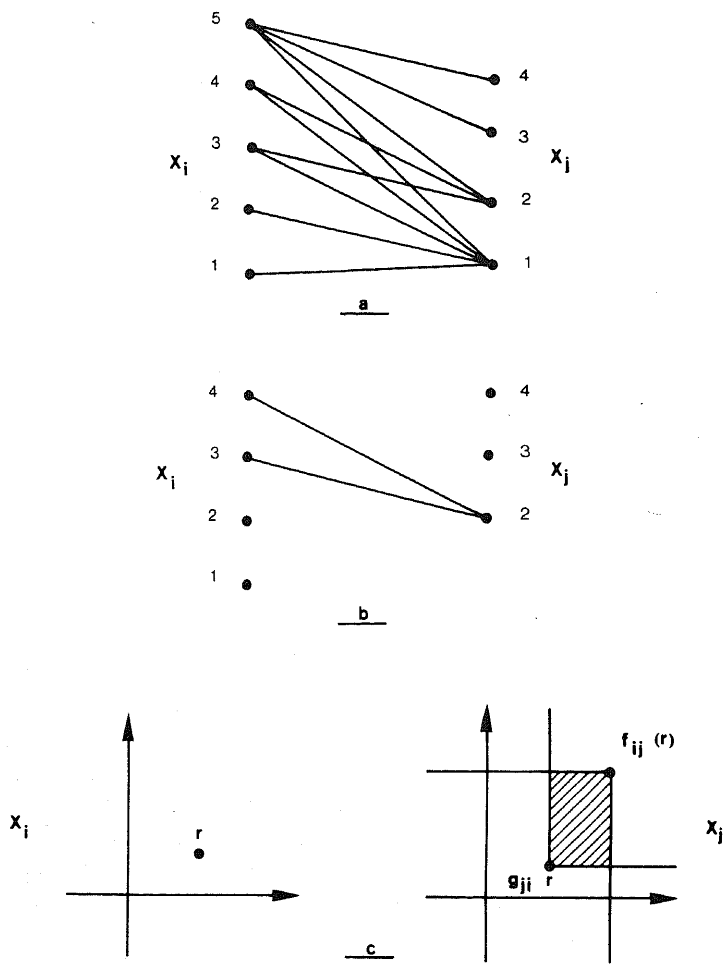

Fig. 6. (a) An example of monotone relation. (b) The relation in (a) without "ceiling" in X_i and "floor" in X_j. (c) The rectangular image of element $x_{i,r}$ as allowed by the intersection of two monotone relations R_{ij} and R_{ji}.

and thus d_{ij} represents the length of the shortest path from vertex V_i to vertex V_j. As a check, note that $d_{ij} \leqslant t_{ij}$ and so $M_{ij} \subseteq R_{ij}$.

If the sets X_i are finite, the restriction to total relations could look heavy. On the contrary, a "floor" value $x_{i,\,0}$ and a "ceiling" value $x_{i,\,u}$ can always be added to X_i, such that

$$R_{ij,\,us} = 1 \text{ for all } s \qquad \text{and} \qquad R_{ji,\,ro} = 1 \text{ for all } r.$$

Fig. 6b shows the relation in Fig. 6a without ceiling in X_i and floor in X_j.

In general, monotone relations are considerably more powerful than simple "shortest path" constraints. For instance, the lattice structure of multidimensional euclidean spaces can be used for specifying multidimensional rectangular domains. In Fig. 6c we see how defining functions f_{ij} of R_{ij} and g_{ji} of R_{ji} restrict to a rectangle the image of $x_{i,r}$ in $R_{ij} + R_{ji}$. Different points $x_{i,r}$ can generate different rectangles, provided equation (5.5) is satisfied.

The next theorem proves the closure of the class of monotone relations under the operations of intersection and composition and gives the rules for performing such operations in terms of the defining functions.

THEOREM 5.5. *(a) If R'_{ij} and R''_{ij} are monotone relations represented by the defining functions f'_{ij} and f''_{ij}, then the sum*

$$R_{ij} = R'_{ij} + R''_{ij}$$

is a monotone relation represented by the defining function

$$f_{ij}(r) = \inf(f'_{ij}(r), f''_{ij}(r)). \tag{5.10}$$

(b) Likewise, the product

$$R_{ij} = R_{ik} R_{kj}$$

is represented by

$$f_{ij}(r) = f_{kj}(f_{ik}(r)). \tag{5.11}$$

Proof. (a) Relation R_{ij} defined by (5.4) and (5.10) is evidently the intersection of R'_{ij} and R''_{ij}. Furthermore R_{ij} is total. In fact we have at least

$$R_{ij,\, r f_{ij}(r)} = 1 \text{ and } R_{ij,\, g_{ij}(s) s} = 1$$

where $g_{ij}(s)$ is defined dually. Finally, function f_{ij} satisfies equation (5.5)

$$\begin{aligned}
f_{ij}(\inf(r_1, r_2)) &= \inf(f'_{ij}(\inf(r_1, r_2)), f''_{ij}(\inf(r_1, r_2)))\\
&= \inf(\inf(f'_{ij}(r_1), f'_{ij}(r_2)), \inf(f''_{ij}(r_1), f''_{ij}(r_2)))\\
&= \inf(\inf(f'_{ij}(r_1), f''_{ij}(r_1)), \inf(f'_{ij}(r_2), f''_{ij}(r_2)))\\
&= \inf(f_{ij}(r_1), f_{ij}(r_2)).
\end{aligned}$$

(b) The "if" part of (5.4) is trivial. For the "only if" part, if $R_{ij,rs} = 1$ then an index t exists, such that $R_{ik,rt} = 1$ and $R_{kj,ts} = 1$. But $R_{ik,rt} = 1$ implies $t \leqslant f_{ik}(r)$. Thus from (5.8) we have

$$f_{kj}(t) \leqslant f_{kj}(f_{ik}(r)) = f_{ij}(r).$$

But $R_{kj,ts} = 1$ implies $s \leqslant f_{kj}(t)$ and thus $s \leqslant f_{ij}(r)$. Relation R_{ij} is also total because at least

$$R_{ij,\, r f_{ij}(r)} = 1 \text{ and } R_{ij,\, g_{ij}(s) s} = 1.$$

Equation (5.5) is proved as follows

$$f_{ij}(\inf(r_1, r_2)) = f_{kj}(f_{ik}(\inf(r_1, r_2)) = f_{kj}(\inf(f_{ik}(r_1), f_{ik}(r_2)))$$
$$= \inf(f_{kj}(f_{ik}(r_1)), f_{kj}(f_{ij}(r_2))) = \inf(f_{ij}(r_1), f_{ij}(r_2))$$

Q.E.D.

Next theorem proves the distributivity of monotone relations.

THEOREM 5.6. (*a*) *Any set of monotone relations*

$$R_{ik}, R_{kj} \quad i, j = 1, \ldots, n$$

form a distributive set of relations with respect to set X_k, *i.e.*,

$$\left(\sum_{i=1}^{p} V_{0i} R_{ik}\right)\left(\sum_{j=1}^{q} R_{kj} V_{j0}\right) = \sum_{i=1}^{p} \sum_{j=1}^{q} V_{0i} R_{ik} R_{kj} V_{j0} \tag{5.12}$$

for all fundamental vectors V_{0i} *and* V_{j0}.

Proof. For Theorem (5.4) vectors $R^i_{0k} = V_{0i} R_{ik}$ and $R^j_{k0} = R_{kj} V_{j0}$ represent subsets of X_k of the form

$$R_{0k,t} = 1 \text{ iff } t \leqslant \bar{t} \text{ and } R_{k0,t} = 1 \text{ iff } t \geqslant \bar{t}. \tag{5.13}$$

Furthermore, intersection of two subsets of this form produces a subset of the same form. If

$$R_{0k} = R'_{0k} + R''_{0k}$$

we have

$$R_{0k,t} = 1 \text{ iff } t \leqslant \inf(\bar{t}', \bar{t}'')$$

And if

$$R_{k0} = R'_{k0} + R''_{k0}$$

we have

$$R_{k0,t} = 1 \text{ iff } t \geqslant \sup(\bar{t}', \bar{t}'').$$

We will prove that subsets of this form satisfy both left and right distributivity

$$R_{0k}(R'_{k0} + R''_{k0}) = R_{0k} R'_{k0} + R_{0k} R''_{k0}. \tag{5.14}$$

$$(R'_{0k} + R''_{0k}) R_{k0} = R'_{0k} R_{k0} + R''_{0k} R_{k0}. \tag{5.15}$$

In fact, the right member of (5.14), in binary form, is

$$\left(\bigvee_{t=1}^{N_k} R_{0k,t} \wedge R'_{k0,t}\right) \wedge \left(\bigvee_{t=1}^{N_k} R_{0k,t} \wedge R''_{k0,t}\right).$$

Applying binary distributivity, we have

$$\bigvee_{t_1=1}^{N_k} \bigvee_{t_2=1}^{N_k} (R_{0k,t_1} \wedge R_{0k,t_2}) \wedge (R'_{k0,t_1} \wedge R''_{k0,t_2})$$

or equivalently

$$\bigvee_{t=1}^{N_k} R_{0k,t} \wedge (R'_{k0,t} \wedge R''_{k0,t}) \vee \bigvee_{t_1=1}^{N_k} \bigvee_{t_2=1}^{N_k} \cdot (R_{0k,t_1} \wedge R_{0k,t_2}) \wedge (R'_{k0,t_1} \wedge R''_{k0,t_2}) \quad (5.16)$$

From (5.13), if

$$R_{0k,t_1} = 1, R_{0k,t_2} = 1, R'_{k0,t_1} = 1 \text{ and } R''_{k0,t_2} = 1$$

then a value

$$t = \sup(t_1, t_2)$$

can be found, such that

$$R_{0k,t} = 1, R'_{k0,t} = 1 \text{ and } R''_{k0,t} = 1.$$

Therefore equation (5.16) becomes

$$\bigvee_{t=1}^{N_k} R_{0k,t} \wedge (R'_{k0,t} \wedge R''_{k0,t}),$$

i.e., the left member of (5.14). Formula (5.15) can be proved dually. From closure under sum, left and right distributivity we have

$$\left(\sum_{i=1}^{p} R^i_{0k}\right)\left(\sum_{j=1}^{q} R^j_{k0}\right) = \sum_{i=1}^{p} \sum_{j=1}^{q} R^i_{0k} R^j_{k0} \qquad \text{Q.E.D.}$$

i.e., formula (5.12).

The next corollary will be useful in establishing star-distributivity.

Corollary. If R_{ik}, R'_{ik}, R''_{ik}, R_{kj}, R'_{kj} and R''_{kj} are monotone relations we have

$$R_{ik}(R'_{kj} + R''_{kj}) = R_{ik}R'_{kj} + R_{ik}R''_{kj} \quad (5.17)$$

$$(R'_{ik} + R''_{ik})R_{kj} = R'_{ik}R_{kj} + R''_{ik}R_{kj}. \quad (5.18)$$

Proof. Equation (5.17) can be written as

$$V_{0i}R_{ik}(R'_{kj}V_{j0} + R''_{kj}V_{j0}) = V_{0i}R_{ik}R'_{kj}V_{j0} + V_{0i}R_{ik}R''_{kj}V_{j0}$$

for all fundamental vectors V_{0i} and V_{j0}. Therefore, it descends from (5.12). The same is true for (5.18). Q.E.D.

We can now prove our final result.

THEOREM 5.7. *Let R be a network of relations such that*
(i) *its relations* R_{ij} $(i, j = 1, \ldots, n; i \neq j)$ *are monotone and*
(ii) *the loop relations* Y^n_{ii} $(i = 1, \ldots, n)$ *of the network* Y^n *obtained after one iteration of algorithm C are equal to unity.*

We can prove that:
(*a*) *Network R is star-distributive.*
(*b*) *Network* Y^n *is equal to the closure Y of R. All relations* Y_{ij} $(i, j = 1, \ldots, n; i \neq j)$ *are monotone.*
(*c*) *Network Y is distributive.*
(*d*) *Network R is decomposable and the symmetrization* Y' *of its closure Y is minimal.*

Proof. (a) From condition (ii) and $Y^n \subseteq R$ we have $R_{ii} = I_{ii}$. Expressions obtained by combining relations R_{ij} $(i, j = 1, \ldots, n; i \neq j)$ with the operations of intersection and composition evaluate to monotone relations for theorem (5.5). The unity elements R_{ii} can be involved in an expression either under composition or under intersection. In the former case a monotone relation is trivially obtained. In the latter case the unity I_{ii} must be intersected with an expression representing the global constraint given by a set of circuits from V_i to V_i in R. In fact we can assume inductively that no unity is involved in this expression, and in this case distributivity holds for the corollary to Theorem (5.6), and the expression can always be reduced to a sum of products. The result of the intersection operation must be again unity, because $Y^n_{ii, rr} = 1$ for all r, and thus for Theorem (4.4) all pairs $(x_{i,r}, x_{i,r})$ must be allowed by all circuits from V_i to V_i in R. In conclusion, the set D of all expressions contains expressions that either evaluate to monotone relations or to identities.
Left distributivity

$$E_{ik}(E'_{kj} + E''_{kj}) = E_{ik}E'_{kj} + E_{ik}E''_{kj} \tag{5.19}$$

always hold. In fact, if all relations are monotone, this property is proved by the corollary to Theorem (5.6). If E_{ik} is a unity, both members evaluate to $E'_{kj} + E''_{kj}$. If E'_{kj} or E''_{kj} is a unity, say E'_{kj}, we have $k = j$ and (5.19) becomes

$$E_{ik}(I_{kk} + E''_{kk}) = E_{ik}I_{kk} + E_{ik}E''_{kk}.$$

But then $E_{kk} = I_{kk}$ for $Y^n_{kk} = I_{kk}$, and therefore both members evaluate to E_{ik}. The same proof holds for right distributivity.
(b) This part follows from (a) and Theorem (5.3).
(c) This part follows from (b) and Theorem (5.6).
(d) This part follows from (c) and Theorem (5.2). Q.E.D.

CONCLUSION

In this paper we have presented a formal treatment of networks of binary constraints. The main practical result was the discovery of an algorithm for adding to the direct constraint between each pair of variables the indirect constraints transmitted by all the paths in the network. In particular cases the resulting constraint was proved equivalent to the global constraint represented by the entire network as seen by that pair of vertices. This result allows the partial or total utilization of the global constraint structure for reducing the set of feasible values of a variable to be determined, when the values of other variables are known.

For the practical computer implementation of this method, the following requirements can be suggested:

(a) In the application under examination, most constraints must be reasonably represented or approximated by binary constraints or simple networks of binary constraints. Note that if we allow a constraint among m variables to be represented by a network of n vertices, with $n > m$, then the negative result of Section 3 no longer holds, and many representations of the constraint, trivial and not, can be found. For instance, the ternary relation (3.4) which is not representable with a 3-vertex network, can be represented by the 4-vertex network in Fig. 2, as seen from vertices V_1, V_2, and V_3.

(b) The resulting binary relations (finite or infinite) must be capable of being stored in an economical way in a computer memory. For instance, if the variables are points of m-dimensional spaces, a relation R_{ij} could be stored representing the images in X_j of all elements[7] $x_{i,r}$ of X_i as m-dimensional domains. Known techniques of domain encoding can then be used. For instance, two given points are sufficient for determining a rectangular domain: this is often the meaning of functions $f_{ij}(r)$ and $g_{ji}(r)$ representing a monotone relation.

(c) The operations of intersection and composition must be easily definable in the chosen class of relations. In particular, this class must be closed under those two operations. For instance, this is the case of relations represented by domains, convex domains, domains enclosed by polygons or convex polygons, rectangular domains.

(d) The closed network is then obtained with algorithm C. The closed network should then be close to the minimal. For instance, we have coincidence for rectangular domains, and we expect reasonable closeness for convex domains. Bad results can be expected if the relations allow most pairs and forbide a few isolated pairs, like in graph-coloring problems. Anyway, if the addition of a further constraint destroys regularity (i.e., closed $\neq$ minimal), it is, nevertheless,

---

[7] Or just one, if all the other images can be obtained from it by a fixed procedure (e.g., translation).

convenient to add it. Maybe its addition will not be entirely exploited, but the monotonicity property of intersection and composition certifies that the modified closed network will be more restrictive.

REFERENCES

1. C. M. Eastman, Representations for space planning, *CACM 13*, 4, 242–250 (April 1970).
2. A. Guzman, Computer recognition of three-dimensional objects in a visual scene, *MAC-TR-59* Project MAC Massachusetts Institute of Technology, December 1968, PhD thesis.
3. M. D. Kelly, Edge detection in pictures by computer using planning, *AIM-108* Stanford Artificial Intelligence Project, Stanford University, January 1970.
4. U. G. Montanari, On the optimal detection of curves in noisy pictures, *CACM 14*, 335–345 (1971).
5. R. Narasimhan, On the description, generation, and recognition of classes of pictures, in *Automatic Interpretation and Classification of Images* A. Grasselli, Ed., Academic Press, New York (1969).
6. M. B. Clowes, Transformational grammars and the organization of pictures, in *Automatic Interpretation and Classification of Images* A. Grasselli, Ed., Academic Press (1969).
7. J. Feder, Linguistic specification and analysis of classes of line patterns, *403-2* Dept of Electrical Engineering, New York University, April 1969, PhD thesis.
8. A. C. Shaw, The formal description and parsing of pictures, *SLAC 84* Stanford Linear Accelerator Center, Stanford University, March 1968, PhD thesis.
9. R. H. Anderson, Syntax-directed recognition of hand-printed two-dimensional mathematics, applied mathematics, Harvard University, 1968, PhD thesis.
10. U. G. Montanari, Separable graphs, planar graphs and web grammars, *Information and Control* 16, 3, 243–267 (May 1970).
11. S. Warshall, A theorem on boolean matrices, *JACM* 9, 11–12 (1962).
12. R. W. Floyd, Algorithm 97 shortest path, *CACM* 5, 6, 345 (June 1962).
13. A. Ginzburg, Algebraic Theory of Automata, Academic Press (1968).
14. J. C. Schwabel, and B. McCormick, Consistent properties of composite formation under a binary relation, *Inform. Sci.* 2, 179–209 (1970).

Received September 22, 1972