ANALYSIS OF CHEMICAL TECHNOLOGICAL NETWORKS

By

F. TÁTRAI, J. BÁRKAI and I. VÁMOS

Department of Chemical Technology, Technical University, Budapest

Received December 5, 1979

Presented by Prof. Dr. I. SZEBÉNYI

Introduction

Technical development in the petroleum processing and petrochemical industry, similarly as in other branches of chemical industry, resulted in the establishment of new complex works of high intensity and high individual performance. The analysis of these typically continuous, integrated chemical-technological systems of continuous operation required a new mode of contemplation: instead of the so called operational unit oriented concept the study of the interaction of the single operational units became the center of interest. The so called "process engineering" concept has been developed. This considers chemical-technological systems as a system consisting of a certain number of equipments (operational units), connected by material and energy streams (in the general sense, by stream vectors).

For the process engineering mapping of chemical-technological systems generally directed graphs, derived from abstractions of various level of the process flow — sheet, are used. Graph theoretical mapping, used also in Hungary [1, 2], proved to be one of the important theoretical and practical methods of process engineering, moreover, the modes of giving the topology of the graphs furnished the basis for the investigation of the features of chemical-technological networks.

Our paper deals with the analysis of process networks representing integrated technological systems, and within this with different methods of the partitioning of process networks.

Partitioning of the process network

The integrated petroleum processing or petrochemical network is characterized by a complicated connection of equipments (operational units). From the known technological connections (series, by-pass, parallel, recycle, crossed [3]) essentially two kinds of sub-systems can be built up: open and closed subsystems. Expediently, open and closed sub-systems will be treated differently

in the calculation of the material and energy balances of process networks. Characteristic of the open sub-system is namely the fact that each technological stream passes only once through whichever element of the system. Thus, following the structural graph and proceeding from element to element, material and energy balance equations can be solved sequentially. Sequential calculation is made possible by the very fact that each process stream passes only once through any element of the system, so that it can not happen that some parameter of the input stream of an element depends on the value of one of its own output streams.

On reaching a closed sub-system, the situation changes, as the closed subsystem is characterized by the very fact that a group of elements is combined by at least one recirculated material or energy stream into a closed cycle, and the input streams of the elements belonging into the closed cycle depend also on the output stream values of the element. Therefore, two solutions are offered for the calculation of the material and energy balances of closed cycles (recycle loops):

a) Tearing of the recycle loops. This is accomplished by taking an arbitrary value for a selected stream as starting value of iteration, and performing sequentially the calculations until the cycle "closes". If the value calculated agrees within the given limits of error with the value assumed, calculation is considered as terminated. If the deviation is larger than the preset tolerance, calculation is continued with the newly calculated starting value (direct iteration) or with a new value obtained from it by some transformation (accelerated iteration processes).

b) Formulating the set of equations of the recycle loops and the simultaneous solving of the set of equations.

If an integrated chemical process network consists of several closed and open sub-systems, expediently the system will be partitioned by the investigation of the structural graph into a sequence of maximal closed sub-systems, which are connected only in one direction with another. A closed sub-system is called maximal, if all the other recycle loops of the net are either contained in it, or are with it in unidirectional (open) connection.

For the identifying of open and closed subgraphs an algorithm is applied.

Identification of maximal closed subsystems is called partitioning. Partitioning in the directed graph means the finding of a set of nodes, $\mathfrak{L}$, belonging to such maximal closed path*, that all other cyclic paths of the graph shall either be totally comprised in $\mathfrak{L}$, or shall have no mutual nodes.

Of the methods of partitioning, the method of powers of adjacency matrix, proposed by NORMAN [4], is of fundamental importance. This method has been

* Cyclic path in the directed graph is the sequence of edges, passing through which from a given starting node one can return again to the same node.

further developed by several authors. KEHAT and SHACHAM [5] simplified the algorithm by replacing the generally sparse (few non-zero elements containing) adjacency matrix for a two-column matrix, in which only the indexes of the non-zero elements are arrayed, and interpreted for this index matrix "logical powers".

In our comparative investigations, programs were prepared for the running of the algorithm based on the powers of the adjacency matrix, of the algorithm based on the powers of the index matrix, and of the algorithm based on direct graph path searching, developed by us. The operation of the program was compared by solving of several examples.

Identifying recycle loops by powers of the adjacency matrix

Adjacency matrix is a form of coding of the directed graph. The number of rows and of columns of the matrix corresponds to the number of nodes of the directed graph. The value of the a_{ij}-th element of adjacency matrix A is 1, if there exists a directed edge (process stream, information stream) from node i to node j. The directed graph and the adjacency matrix of a part of the technological process is shown in Fig. 1.

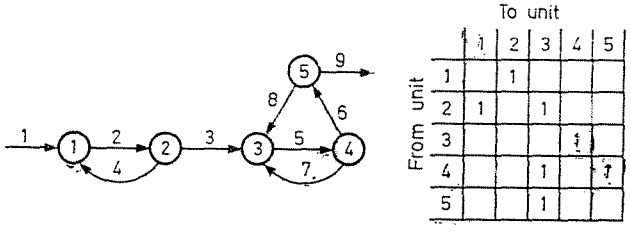


Fig. 1. Directed graph and adjacency matrix of a part of the technological process

The algorithm for the identifying of maximal recycle loops is the following:

Let us examine the adjacency matrix, whether it has columns or rows containing only 0 elements. If one column of the matrix obtained contains only 0 elements, this means that no stream enters the unit at the "head" of the given column from a unit of higher series number, therefore the given unit can not be part of a recycle loop. Removing the unit from the process (i.e. delating its row and column from the matrix), the new matrix either does not contain anymore only 0 columns, or such column is still present. Successively these units are also eliminated, and their number is placed in a list. Next it is examined on a similar principle, whether there is a row in the matrix consisting of only 0 elements, i.e. whether there is a unit, from which no process stream enters in any of the other process units. These cases are also eliminated, and the algorithm is continued again by the examination of the columns. The algorithm is termi-

nated in finite steps, and the residual matrix contains only the numbers of units, which are contained in the single recycle loops or are located inserted between two recycle loops. In our case, the result does not differ from the original matrix.

Let us put to powers this matrix by Boolean (logical) arithmetic:

The a_{ij}-th element of the first power of the matrix is 1, if stream flows from unit i to unit j, and is 0 if not, or in another wording, a path consisting of a single edge leads from node i of the graph to node j. The a_{ij}-th element of the second power, A^2, is 1, if there exists in the graph a path consisting of two edges from node i to node j. Marking the sum of the first and second matrices with R_2:

$$R_2 = A + A^2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \quad (1)$$

In the process a unit is part of a recycle loop, if there exists in the directed graph a cyclic path consisting of an arbitrary number of edges from unit i to unit i, that is to say, on introducing the following notation:

$$R_n = \sum_{i=1}^{n} A^i; \quad R_\infty = R = \lim_{n \to \infty} R_n, \quad (2)$$

along the main diagonal of R, 1 stands at the place of the given element. In our case:

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & 1 & 1 & 1 \end{bmatrix}. \quad (3)$$

If two units are contained in the same recycle loop, feed stream flows both from unit i to unit j, and from unit j to i, possibly through several units. Since matrix R expressed whether there exists in the graph a set of edges from node i to node j, its transpose R^T, will express whether there is a path from unit j to unit i in the graph. If both paths exist, the said units belong to the same recycle loop, which can be expressed that in the matrix

$$W = R \cap R^T \quad (4)$$

both the (i, j)-th and the (j, i)-th elements are 1. For the parts of the process investigated

$$W = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & & & \\ 1 & 1 & & & \\ & & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & 1 & 1 & 1 \end{bmatrix}. \tag{5}$$

This is consistent with Fig. 1, where units 1 and 2, and 3, 4 and 5 are contained in each of a recycle loop.

As concerns the criticism of the method, on changing the numeration of the units (the nodes of the graph), the expressive block structure of the matrix disappears. In complicated networks numeration along the direction ("in downstream direction") is not always possible. In these cases, the expressive block structure can be generated by transformation with the appropriate permutation matrix. The algorithm can be most advantageously realized on computers programable also for binary arithmetic operations.

Partitioning on the basis of the powers of the index matrix

The adjacency matrix is rather sparce, that is to say, it contains many 0 elements, so that to save computer memory expediently only the list of the indexes of the non-zero elements will be stored.

The idea presented itself that the search for maximal recycle loops, i.e. partitioning, can be performed by the power raising of only the index matrix [5]. (This does not mean true power raising, because the power of the index values would have no physical content, but it would mean only a series of logical operations for the generation of index pairs of the non-zero elements of the A matrix powers.) This can result in the given case a substantial saving in memory, because the adjacency matrix contains in the case of m units and n streams m^2 elements, while the index matrix only $2\ n$ elements.

The steps of the algorithm are:

a) The feed streams and the product streams of the system are eliminated (those streams which come from or go to the fictive 0 node).

b) Nodes without input stream are also eliminated, and are placed on a forward list.

c) Nodes without output stream are olso eliminated, and placed in a backward list.

d) Steps 1 and 2 are repeated, till no more nodes without input or output are found.

e) The I matrix is raised to powers: the right hand element of each coordinate pair (which is also the number of a node) is replaced by the right hand

element belonging to the coordinate in the left hand column of matrix I, to the given node. (Thus, precisely the coordinates of the elements in the powers of adjacency matrix A get into the rows of power of matrix I.)

The index matrix of the directed graph relevant to the part of the technological process shown in Fig. 1, and the first and second powers of the index matrix are contained in Table 1

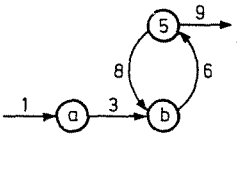
Table 1

Streams	I	I'	I'²
1	0 1		
2	1 2	1 2	1 3
3	2 3	2 3	1 1
4	2 1	2 1	2 4
5	3 4	3 4	2 2
6	4 5	4 5	3 5
7	4 3	4 3	3 3
8	5 3	5 3	4 3
9	5 0		4 4
			5 4

f) In the p-th power, recycle loops consisting of p streams (indicated in the diagonal of the adjacency matrix by 1) are characterized by the fact that the right hand and left hand coordinates of the given streams are identical.

g) If the number of identical pairs is greater than p, one of the pairs (one of the nodes) is delated and the algorithm is repeated from step *b)*. The new I^p matrix is compared with the earlier, and identical pairs now missing from matrix I^p are combined into a single pseudo-node, and step *g)* is repeated. In our example 4 identical pairs are in the second power, therefore, one node e.g. 1 is eliminated. Thereby 2 is also delated in the list. Raising to power of the residual matrix, identical pairs of nodes 1 and 2 are missing from the 2nd power, therefore these can be combined into a pseudo-node marked with "a".

h) If the number of identical pairs in the p-th power is just p, all identical pairs can be combined into a single pseudo-node, and the process is repeated from step *b)*. The algorithm ends, when no more nodes remain after step *d)*. In the example shown in Fig. 1, there are just 2 identical pairs in the 2nd power after the combination of nodes 1 and 2, these are combined as pseudo-node of marking "*b*". The directed graph and its index matrix obtained in this way are shown in Fig. 2.



	I	
1	0	a
3	a	b
6	b	5
8	5	b
9	5	0

Fig. 2. The directed graph shown in Fig. 1 and its index matrix after the combination of node pairs (1, 2) and (3, 4)

Repeating the algorithm for index matrix in Fig. 2, the second power consists now only of the identical pairs *(b,b)* and (5,5). On combining these into a single node "*c*", the open graph obtained will be the following (Fig. 3):

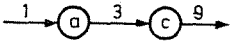


Fig. 3. The open graph obtained from the directed graph of Fig. 1

The index matrix of this graph is exhausted already in steps *a.*, and *b.*, of the algorithm, and the algorithm ends.

Partitioning by path searching along the directed graph

The first algorithm of this kind was published by Sargent and Westerberg. According to the algorithm the graph is traced, and nodes which have been already encountered are listed. If a node is encountered again, this indicates a recycle loop. The Loopfinder algorithm of FORDER and HUTCHINSON is a further development of this algorithm [6, 7].

The algorithm described in the following is essentially based on the same principle, that is to say, it systematically traces the streams of the network. This algorithm is in so far an improvement over those known so far, as path searching and coding of the recycle loops found are simpler and more rapid.

Let us investigate the operation of the algorithm on the part of the chemical process shown in Fig. 1.

In tracing the graph, the algorithm operates with the ordered index matrix. Ordering is done according to the first column of the index matrix (according to the node from which the given directed edge starts), arranging the streams in a not strictly monoton increasing sequence.

Let the graph investigated be the same as that shown in Fig. 1, and its index matrix shall be identical with the first index matrix in Table 1. Two "working vectors" are taken. For the sake of descriptiveness we will call the first "stack", because the listing and delating of elements is always undertaken in

the opposite order to filling. (Information stored in the "stack" becomes accessible only when that is taken out first, which is at the "top".) For the algorithm the number of streams (directed edges) must be given, and as starting information the edge with the examination of which path searching is to begin. Initially both the stack and the vector serving for the storage of the recycle loops already found are empty.

Let us begin path searching with stream 1 ($K = 1$). According to the testimony of the second column of the index matrix, stream 1 enters unit 1 ($I(K, 2) = 1$). It can be seen from the graph of the network (Fig. 1) that operational unit 1 has 1 output stream, stream 2. This must be found by the algorithm. The procedure is to examine backwards along the first column of the index matrix, starting from stream 9, whether I(J,1) is identical with I(K,2), i.e. whether operational unit 1 has at all an output stream. On finding stream 2, this will be the new stream to be examined, and the first stream 1 will be placed in the stack.

With the same steps stream 8 is reached, placing in the stack streams 2, 3, 5, 6, 8. In the following figure (Fig. 4) the dotted line marks the path of tracing.

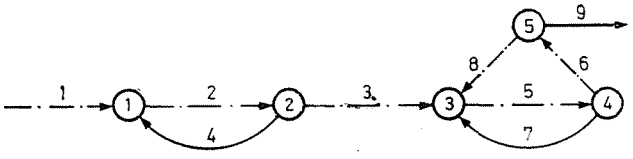


Fig. 4

On arriving to stream 8, inlet to unit 3, unit 3 has only one output stream, which has been already encountered, stream 5. Stream 5 is not written again in the stack, but streams which have been placed in the stack up to stream 5 (5-6-8) are read out from the stack. This is written as a recycle loop found in the vector used for the storage of the recycle loops (this is the first loop found, there can be no question of its repeated finding).

The last unit examined was unit 3. This has no more output stream, and neither is the stack empty. (Streams for the present in the stack are 1, 2, 3, 5, 6, 8.) The new stream examined will be stream 8, stored at the top of the stack. This is taken out from the stack (is delated). Output stream 8 comes from operational unit 5, the next output stream of which is stream 9. Since this stream goes to the environment, the tracing of the network "dies away" along this branch (Fig. 5).

We turn now again to the streams stored in the stack. Stream 6 is located "at the top" of the stack. This will be the next stream examined (at the same time it is delated from the stack; the state of the stack is now 1, 2, 3, 5).

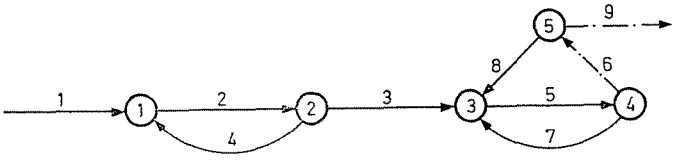


Fig. 5

Stream 6 is the output of unit 4. Since this unit has a further output stream, stream 7, this will be the next stream examined. Tracing along stream 7, we arrive back to the already traced stream 5, finding thus another recycle loop (Fig. 6).

Since this loop has not yet been encountered, it is written as second recycle loop into the vector storing the recycle loops.

We are now again at unit 3, from where we would find again according to the algorithm the recycle loop 5—6—8, but this is "spotted" by the sub-algorithm serving for the ordering of the loops, and it steps on. Streams 5 and 7 are delated from the stack, and arriving now at stream 3, we get to operational unit 2, the output stream of which has not yet been examined, and this will be the stream actually examined. The last steps of the tracing of the network are shown in Fig. 7.

After finding of the recycle loop consisting of streams 2—4 and their delation from the stack only stream 1 remains in the stack, which is also delated (since the output stream of unit 1 has been already examined). The stack becomes empty, the tracing of the network is ended. Thereby, all the recycle loops of the network have been identified. The identifying of the maximal recycle loops is performed by the algorithm working by the comparison of the streams of the recycle loops.

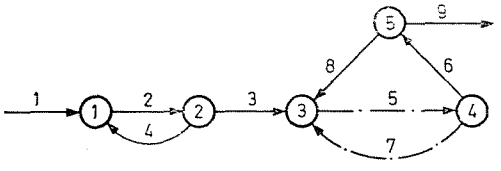


Fig. 6

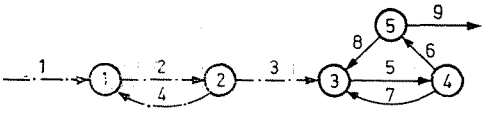


Fig. 7

The fact that the algorithm performs the identifying of both the total and the maximal recycle loops involves further advantages. It is known namely that in addition to structural or information graphs constructed on the basis of the process graph, the simulation of chemical process networks can be undertaken also by the drawing of the so called signal flow graph [9, 10]. The edges of the signal flow graph are the nodes of the original graph, its nodes the edges of the original graph. In the case of chemical process graphs this means that the nodes of the signal flow graph are the streams of the process, while its edges represent the transformation of the streams produced by the operational units. Signal flow graphs are widely used for the simulation of chemical engineering systems, if the mathematical model of the operational unit can be written in the linear form [3, 9]. In this case the signal flow graph is drawn, and this is followed by the identifying of all the recycle loops and the so called forward paths (sets of edges free of cycle, belonging to and preceding the stream to be calculated). When these are known, the explicite formula of the stream searched can be written as a function of the input streams and the transformations produced by the operational units. This is the so called Mason's rule.

One of the advantageous properties of the "stack treating" algroithm developed by us is the very fact that both in simulation based on the structural (information) graph and in simulation based on the drawing of the signal flow graph it can be incorporated into the program segment performing the analysis of the network. This is shown in Fig. 8.

A further advantage of the algorithm is to perform but few "superfluous" operations, so that on an identical computer, operated in an identical operational system, its running time, in the case of programs of running time optimized in the same measure, is shorter than those of the programs written on the basis of the two algorithms discussed earlier. This was proved by trial runs on

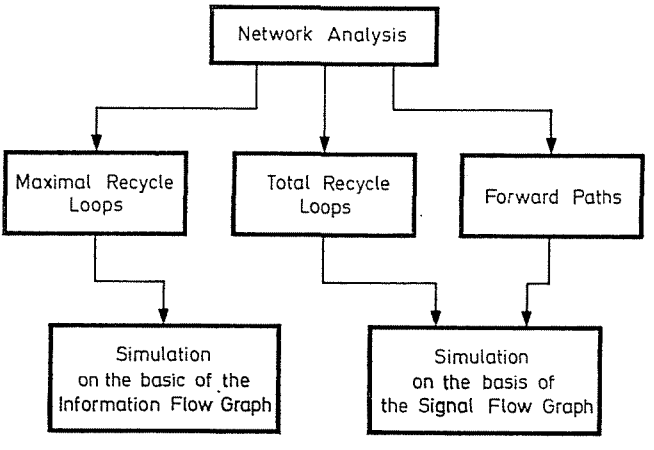


Fig. 8

a small computer WANG Model 2200, programmable in BASIC language. The following figure shows the graphs used for the two trial-runs, and the approximative running times needed for the finding of the maximal recycle loops.

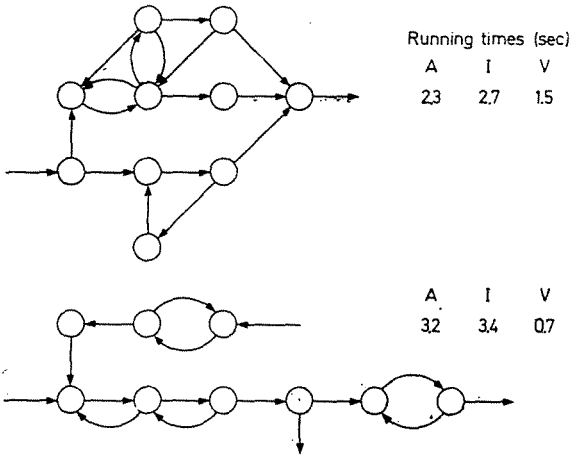


Fig. 9

Running time obtained for the powered adjacency matrix is marked with A, that for the powered index matrix with I, and running time obtained by the application of the stack treating technique with V.

Summary

Our paper deals with the analysis of process networks representing integrated technological systems, and within this with different methods of partitioning of the process network. Partitioning means the identification of maximal closed subgraphs in a directed graph. Of the methods of partitioning, the method of powers of the adjacency matrix, proposed by Norman, the method of powers of the index-matrix, proposed by Kehat and Shasham, and a new path searching algorithm, developed by us, are compared.

References

1. Korach, M.: Operation, process, procedure. Lecture held at the Conference of Intensive Chemical Processes, Kecskemét 1964
2. Korach, M.—Haskó, L.: Graph-theoretical mapping of chemical engineering processes. Akadémiai Kiadó, Budapest, 1975
3. Kafarov, V. V.—Perov, V. L.—Mesalkin, V. P.: Mathematical modeling of chemical engineering systems. Műszaki Könyvkiadó Budapest 1977
4. Norman, R. L.: A matrix method for location of cycles in a directed graph. Am. Inst. Chem. Eng. J. **11**, 450 (1965)
5. Kehat, E.—Shacham, M.: Chemical process simulation Programs-1, Process Technol. **18**, (1/2) 35 (1973)
6. Sargent, R. W.—Westerberg, A. W.: Trans. Inst. Chem. Eng. **42**, 190 (1964)

7. FORDER, G. J.—HUTCHINSON, H. P.: Chem. Eng. Sci. **23,** 771 (1969)
8. VÁMOS, I.: Analysis of chemical process networks by path searching methods. Diploma work. Department of Chemical Engineering of the Technical University of Budapest, 1979
9. HIMMELBLAU, D M.—BISCHOFF, K. B.: Process analysis and simulation: Deterministic systems, Wiley. New York. London, Sidney 1968
10. MASON, S. I.—ZIMMERMANN, H. J.: Electronic circuits, signals and systems. J. Wiley. New York 1960

Dr. Ferenc TÁTRAI, János BÁRKAI H-1521 Budapest

István VÁMOS, FŐMTERV, H-1014 Budapest, Uri u. 64/66.