STUDY OF INTEGRATED USE OF SIMULATION ALGORITHM IN OPTIMIZATION OF CHEMICAL PROCESSES

J.KLEMEŠ*, M.DOHNAL and V.VAŠEK*

Department of Chemical Engineering,
Faculty of Mechanical Engineering, Technical University, 600 00 Brno

Received February 17th, 1972

An algorithm for calculation of optimum design variables of unit operations in chemical processes is proposed, based on a simulation programming system. It has all advantages of universal simulation algorithms: a possibility to solve processes with different topology without a change of the program, including recycles, and with various operations. Instead of a direct change of design variables, the quantities are changed that are used in balance equations of the simulation algorithm. The problem can thus be linearized and solved by linear programming methods with regard to constraints involved in the balance equations. It is, however, necessary to recalculate the design variables of unit operations from the balance values. The Nagiev approach is used for simulation; the calculation can be modified even for sequential methods. The proposed optimization method was used previously in solving simple problems.

Optimization of chemical processes is an important task from the economical point of view. Its solution involves many theoretical and practical difficulties. Optimization of a chemical process in a steady state can be treated as a problem of mathematical programming[1]:

$$Z = z(\boldsymbol{X}), \qquad \boldsymbol{\Phi}(\boldsymbol{X}) \geqq \boldsymbol{0}, \qquad (1), (2)$$

where $\boldsymbol{X}$ denotes n-dimensional vector of independent variables, $z(\boldsymbol{X})$ objective function and $\boldsymbol{\Phi}(\boldsymbol{X})$ v-dimensional vector of constraints which must be satisfied by the sought values of the vector $\boldsymbol{X}$ of independent variables that extremize the objective function $z(\boldsymbol{X})$.

When the flow sheet of the process is known (*i.e.* the mode of mutual connection and types of unit operations), the independent variables in optimization of chemical processes are design and "operation" parameters (*e.g.* heat-exchange surfaces, volumes, and pressures). The number of independent variables is high even for relatively simple processes. Constraints are given by balance equations, mathematical models of apparatuses, and some other relations. In using the mathematical programming

* Present address: Department of Applied Mathematics, Research Institute of Chemical Equipment, Brno.

methods, it is necessary to construct the objective function (*1*) and constraints (*2*) based on the knowledge of a particular process, which all is time-consuming. Every change in mutual connection of the apparatuses or in the type of unit operations namely results in a change of the objective function and of the constraints, which must be formulated anew.

General-purpose simulation programs make it possible to formulate a number of problems of mathematical modelling of technological systems[2,3]. It is therefore advantageous to use these programs in constructing the optimization algorithms. Their universality is helpful especially when the economical suitability of a series of modifications of a particular process is to be checked[4], *e.g.* heat exchange systems or the number of apparatuses in parallel.

Analysis of the problem

For an optimization algorithm it is possible to adopt mathematical models (process and economical) of unit operations as well as algorithms for calculation of physico-chemical properties of components[3,5].

A trivial optimization can be performed by repeated solution of the simulation problem without the use of further mathematical aids. The simulation algorithm is used only in calculating the objective function, and the influence of "manual" changes of the design variables or changes in the topology of the system is studied. An advantage of such a stepwise approximation and study is the possibility of using the simulation system without a further modification; a disadvantage is the dependence on the experience of the designer, impossibility of automation and in most cases a longer time required for solution. After every modification the computation must be stopped, evaluated, and a further program must be specified. Although the consumption of the computer time can thus be minimized if the user is an experienced designer, the tendency nowadays is to automate the work as much as possible, hence to shorten the time spent by the user.

It is advantageous to combine the simulation algorithm with one of the deterministic optimization methods[6], *e.g.* a gradient method, existing in many modifications. A simplified block diagram is shown in Fig. 1. An objective, exhaustive comparison of these methods has not been made hitherto mainly for the reason that the optimization problems differ considerably from case to case. It is a common feature of all these methods that they use the simulation algorithm only in calculating the value of the objective function for a certain state of the optimized process given by specified design variables.

Basic Concepts

Combination of the simulation algorithm with deterministic optimization methods may be more advantageous than described above: The quantities involved directly

in balance equations are changed; the problem can then be simply linearized and solved by linear programming methods. Thus, the balance equations can be used as constraints.

The description of an integrated use of the simulation algorithm must be based on a detailed analysis of the method used in the simulation. The simulation calculation

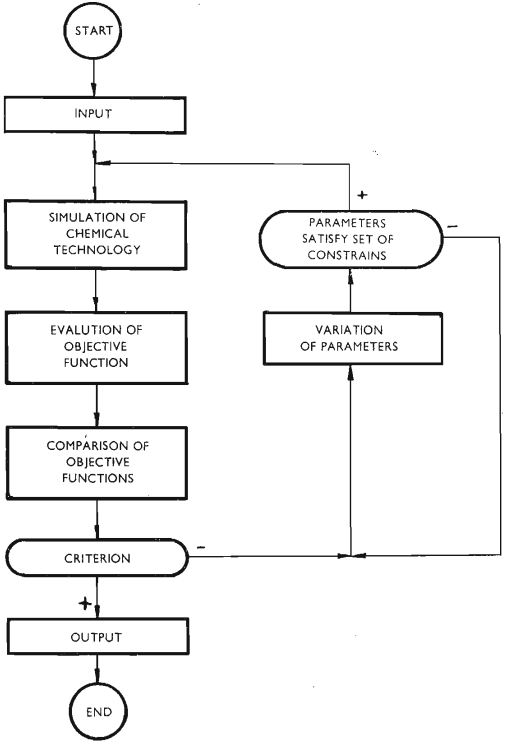

FIG. 1
Simplified Block Diagram

is made either by sequential or simultaneous methods. The integrated use is elaborated in the present work for the simultaneous approach based on the work of Nagiev[7] and Rosen[8].

The basic terms (stream, node, component, design variable, *etc.*) were defined previously[5]. We shall consider an optimized system with K nodes, M streams, and F components. To a stream coming out from the i-th node and entering the j-th node a specification vector, $\boldsymbol{S}_{i,j}$, is assigned, the components of which are the flow rates of all components and the heat flow. The component $S_{i,j}^{[d]}$ is the flow rate of d-th component in a stream leading from i-th to j-th node. The heat flow is considered as $(F + 1)$-st component and the surroundings as zeroth node. The total flow of d-th component through j-th node is given as

$$\lambda_j^{[d]} = \sum_{i=0}^{K} S_{i,j}^{[d]} . \tag{3}$$

The splitting fraction, a, defined by Nagiev[7], for d-th component in a stream coming out from j-th node and entering r-th node is given as

$$a_{j,r}^{[d]} = S_{j,r}^{[d]} / \lambda_j^{[d]} . \tag{4}$$

Vector $\boldsymbol{a}_{i,j}$ with components $a_{i,j}^{[1]}, a_{i,j}^{[2]}, \ldots, a_{i,j}^{[F+1]}$ corresponds to the flow in the stream connecting the i-th and j-th nodes. We introduce matrix $\boldsymbol{A}$, the elements of which are vectors $\boldsymbol{a}_{i,j}$ $(i, j = 1, 2, \ldots, K)$. Vector $\boldsymbol{P}_j$, the components of which are design variables of the unit operation represented by the j-th node, is assigned to each node. The following balance equation for d-th component and j-th node[7] can be derived with the aid of Eqs (3) and (4):

$$\lambda_j^{[d]}(1 - a_{j,j}^{[d]}) - \sum_{\substack{i=1 \\ i \neq j}}^{K} \lambda_i^{[d]} a_{i,j}^{[d]} = S_{0,j}^{[d]} . \tag{5}$$

This equation written for all K nodes $(j = 1, 2, \ldots, K)$ represents a system of linear equations for the d-th component with $\lambda_j^{[d]}$ as unknowns. The splitting fractions $a_{i,j}^{[d]}$ and inlet feeds of the raw material, $S_{0,j}^{[d]}$, are known input data file. If this system is solved in turn with respect to all $F + 1$ components, the values of all components of the stream specification vectors are obtained.

It was assumed in the preceding text that the splitting fractions of the output streams, $\boldsymbol{a}_{i,j}$, for all components and heat are known, hence $\boldsymbol{a}_{i,j}$ depend only on vector $\boldsymbol{P}_i$ of parameters of i-th node:

$$a_{i,j}^{[d]} = f_{i,j,d}(\boldsymbol{P}_i) , \quad j = 1, 2, \ldots, K , \quad d = 1, 2, \ldots, F + 1 . \tag{6}$$

An example of such behaviour is a mixer (Fig. 2, node 4). The splitting fractions of output streams of many other node types depend, however, also on the components of specification vectors of streams which enter the node. The simulation computation must therefore proceed through stepwise approximation of the initial estimate (*e.g.* of the splitting fractions) and the change of stream specification vectors is checked[9]. It is suitable and often necessary to speed up the convergence of the calculation by modified numerical methods[10].

The objective function can be generally expressed in terms of design variable vectors, material prices, necessary energies, and auxiliary materials. We assume for simplicity that the objective function depends only on the vectors of design variables:

$$Z = z(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_K) . \tag{7}$$

The splitting fractions, $a_{i,j}^{[d]}$, are generally functions of the specification vectors of streams entering i-th node and of the design variable vector of node $\boldsymbol{P}_i$:

$$a_{i,j}^{[d]} = f_{i,j,d}(\boldsymbol{P}_i, \boldsymbol{S}_{1,i}, \boldsymbol{S}_{2,i}, \ldots, \boldsymbol{S}_{K,i}) . \tag{8}$$

We assume that vectors $\boldsymbol{S}_{1,i}, \boldsymbol{S}_{2,i}, \ldots, \boldsymbol{S}_{K,i}$ are constant and it is possible to find an inverse function

$$\boldsymbol{P}_i = f_i^{-1}(a_{i,j}^{[d]}, \boldsymbol{S}_{1,i}, \boldsymbol{S}_{2,i}, \ldots, \boldsymbol{S}_{K,i}) . \tag{9}$$

Optimization Algorithm

The block diagram of the optimization algorithm is analogous to the gradient (or direct search) method, Fig. 1. However, the internal structure of the design variable change block is principally different.

The objective function depends on the design variable vectors (7), which can be in a "small" region expressed as functions of the splitting fractions (9). The objective function can be written as

$$Z = z(\boldsymbol{a}_{1,1}, \boldsymbol{a}_{1,2}, \ldots, \boldsymbol{a}_{1,K}, \boldsymbol{a}_{2,1}, \ldots, \boldsymbol{a}_{K,K}) , \tag{10a}$$

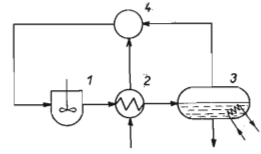

FIG. 2
A Chemical Process

or in an abbreviated form

$$Z = z(\boldsymbol{A}), \tag{10b}$$

and expanded in Taylor series in the point of the initial estimate, $\{\boldsymbol{A}\}$, of the elements of matrix $\boldsymbol{A}$. By neglecting the second- and higher-order terms the equation can be linearized:

$$Z = z(\{\boldsymbol{A}\}) + \frac{\partial z(\{\boldsymbol{A}\})}{\partial a_{1,1}^{[1]}}(a_{1,1}^{[1]} - \{a_{1,1}^{[1]}\}) +$$

$$+ \frac{\partial z(\{\boldsymbol{A}\})}{\partial a_{1,1}^{[2]}}(a_{1,1}^{[2]} - \{a_{1,1}^{[2]}\}) + \ldots + \frac{\partial z(\{\boldsymbol{A}\})}{\partial a_{K,K}^{[F+1]}}(a_{K,K}^{[F+1]} - \{a_{K,K}^{[F+1]}\}). \tag{11}$$

The zero-order term in this equation influences only the absolute value of the extremum. In searching the optimum it can be assumed that

$$z(\{\boldsymbol{A}\}) - \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{d=1}^{F+1}\frac{\partial z(\{\boldsymbol{A}\})}{\partial a_{i,j}^{[d]}}\{a_{i,j}^{[d]}\} = 0. \tag{12}$$

Eq. (*11*) can then be simplified to

$$Z = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{d=1}^{F+1}\frac{\partial z(\{\boldsymbol{A}\})}{\partial a_{i,j}^{[d]}}a_{i,j}^{[d]}. \tag{13}$$

Here the majority of the splitting fractions are equal to zero. The assumption in the general derivation that streams lead from the nodes to all other K nodes of the system is not fulfilled now.

After having expressed the objective function Z as a function of the matrix $\boldsymbol{A}$ and linearized it in the region close to the supposed optimum, we search a matrix $\boldsymbol{A}$ that corresponds to the extremum of Z. Provided that small changes of the splitting fractions, $\boldsymbol{a}$, result in negligible changes of the component flows, λ, through the nodes each balance equation can be approximated by two inequalities:

$$\lambda_i^{[d]}(1 - a_{i,i}^{[d]}) - \sum_{j\neq 1}^{K}\lambda_j^{[d]}a_{j,i}^{[d]} \leqq S_{0,i}^{[d]}(1 + \gamma), \tag{14a}$$

$$\sum_{j\neq 1}^{K}\lambda_j^{[d]}a_{j,i}^{[d]} - \lambda_i^{[d]}(1 - a_{i,i}^{[d]}) \leqq -S_{0,i}^{[d]}(1 - \gamma), \tag{14b}$$

where γ is a chosen positive number. The splitting fractions must fulfil, besides the constraints (*14a,b*), also conditions following from the definition of the splitting

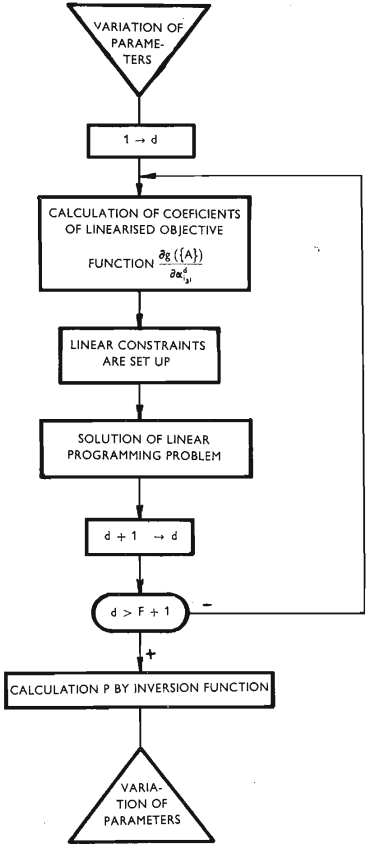

FIG. 3
Block Diagram

fractions:

$$\sum_{j=1}^{K} a_{i,j}^{[d]} = 1, \quad i = 1, 2, \ldots, K, \quad d = 1, 2, \ldots, F + 1. \qquad (15)$$

The linearized optimization problem can now be defined as follows: The splitting fractions that extremize the linear objective function (*13*) and fulfil the linear equations (*15*) as well as the inequalities (*14a,b*) are to be determined. This can be achieved by linear programming methods[11]. If there are M streams in the diagram then the number of nonzero elements of matrix $\boldsymbol{A}$ does not exceed $(F + 1)\,M$. Since the complexity of the numerical solution increases rapidly with the number of unknowns it is advantageous for the linearized problem (*13*)–(*15*) to make use of the principle of superposition of each component and heat[12]. This makes it possible to find the optimum values of the components and heat in the course of one iteration independently, *i.e.* the linear programming problem with M variables is solved repeatedly. From the calculated splitting fractions the corresponding vectors $\boldsymbol{P}_1$ through $\boldsymbol{P}_K$ are calculated with the aid of inverse functions of the type (*9*). The whole procedure is illustrated by the block diagram in Fig. 3.

The described algorithm reduces the nonlinear optimization problem to repeated solution of a linear programming problem which, however, neglects the nonlinear relations involved in the unit operation models. Therefore, the nonlinear model is approximated with a sufficient accuracy only in a certain region close to the point $\boldsymbol{A}$, in which the equations were linearized. In this point the fulfilment of all constraints including nonlinear ones is ensured by the preceding simulation. Consequently, no large changes of the design variables in the course of one iteration are allowed. Their change is limited by the value of γ which must be sufficiently small.

The described method does not ensure always the localization of the global extremum. In practice, however, even the determination of the local extremum or improvement of the initial estimate of design variables is important.

The proposed algorithm makes it possible to linearize the problem of finding optimum values of design variables of unit operations of chemical processes. Balance equations are used as constraints in solving the problem by linear programming methods. Since the simulation calculation is based on the Nagiev approach, the optimization algorithm reflects its advantages and disadvantages. The suitability study of the described method for the treatment of complex real processes, problems of convergence, and derivation of an analogous algorithm based on a sequential simulation approach will be the subject of a further work.

LIST OF SYMBOLS

$\boldsymbol{A}$ matrix of all splitting fractions
$\{\boldsymbol{A}\}$ estimated value of matrix $\boldsymbol{A}$
$\boldsymbol{a}_{i,j}$ vector of splitting fractions of flow in a stream leading from i-th to j-th node
$\{\boldsymbol{a}_{i,j}\}$ estimated value of vector $\boldsymbol{a}_{i,j}$
$a_{i,j}^{[d]}$ component of vector $\boldsymbol{a}_{i,j}$ — splitting fractions for d-th component
F number of mass components
K number of nodes
M number of streams
$\boldsymbol{P}_i$ vector of design variables of unit operation of i-th node
$\boldsymbol{S}_{i,j}$ specification vector in a stream leading from i-th to j-th node
$S_{i,j}^{[d]}$ component of vector $\boldsymbol{S}_{i,j}$ — flow of d-th component
$\boldsymbol{X}$ vector of independent variables
Z objective function
z function
γ chosen positive number
$\boldsymbol{\Phi}$ vector of constraints
$\lambda_j^{[d]}$ total flow rate of d-th component through j-th node

REFERENCES

1. Ostrovskij G. M., Volin J. M.: *Metody Optimizacii Složnych Chimikotechnologičeskich Schem.* Chimija, Moscow 1970.
2. Vermouzek Č.: Die Schwerindustrie der Tschechoslowakei *4*, 6 (1971).
3. Chai T. P., Myles A. F.: General Flowsheet Program, Comp. Progr. No PO 404. M. W. Kellog, New York.
4. Davies C., Perris F. A.: Paper Presented at Internat. Federation of Automatic Control Symposium *Digital Simulation of Continuous Processes*, Györ, September 1971.
5. Dohnal M., Klemeš J.: Chem. průmysl *21*, 345 (1971).
6. Andrew S. M.: Trans. Inst. Chem. Eng. *47*, T 79 (1969).
7. Nagiev M. F.: *Učenije o Recirkulacionnych Processach v Chimičeskoj Technologii.* Azernešr, Baku 1965.
8. Rosen E. M.: Chem. Eng. Progr. *58* (10), 69 (1962).
9. Doležalík V., Vermouzek Č., Dohnal M., Vašek V., Klemeš J.: *Simulace složitých chemicko-inženýrských systémů.* Knižnice odborných a vědeckých spisů VUT Brno, Vol. B-19, Brno 1971.
10. Cavett R. H.: Proc. Am. Petrol. Inst. *43*, III, 57 (1963).
11. Salazar R. C., Subrata K. S.: Comm. of the Assoc. Comp. Machin. *11*, 437 (1968).
12. Dohnal M.: *Thesis.* Technical University, Brno 1971.

Translated by K. Micka.